VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

KONTROLNÍ A ŘÍDICÍ MODUL S IOT
IOT CONTROL MODULE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Martin Haman**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Zoltán Szabó, Ph.D.**
SUPERVISOR

BRNO 2018

ABSTRAKT

Tato bakalářská práce se věnuje průzkumu možností bezdrátového přenosu dat z řídících modulů IoT a jejich konstrukcí. Nejdříve se zabývá porovnáním bezdrátových technologií, síťových protokolů a jejich vhodnosti v závislosti na aplikaci řídícího modulu IoT. Pak je zpracován samotný návrh řídícího modulu IoT s využitím WiFi modulu, založeným na integrovaném obvodu Espressif ESP8266, komunikující pomocí síťového protokolu MQTT. Dále se práce zabývá výběrem, kompilací firmwaru a vytvoření skriptu na řízení vytápění pro řídící modul IoT. Následně jsou popsány způsoby ovládání řídícího modulu IoT a zobrazení naměřených dat. Poté jsou navrženy možnosti vytvoření MQTT broakeru a zabezpečení přenášených dat. Na závěr je zdokumentována samotná realizace řídícího modulu IoT a ověření jeho funkčnosti.

KLÍČOVÁ SLOVA

ESP8266, ESP-12F, NodeMCU, IoT, Wi-Fi, LoRa, Z-wave, Sigfox, ZigBee, HTTP, CoAP, MQTT, řídící modul

ABSTRACT

This bachelor thesis deals with the research of the possibilities of wireless data transmission from the controllers IoT modules and their construction. At first there is compared wireless technologies, network protocols, their suitability depending on the application of the IoT control module. Then it is designed the IoT control module using a Wi-Fi module (based on Espressif ESP8266 integrated circuit) communicating over a MQTT network protocol. Thesis continue with selection, compilation of the firmware, creation of script for control heating system, descriptions of methods for controlling the IoT control module and display of measured data. The possibilities of creating a MQTT broker and security of data transmission are described below. In conclusion is documented and verified its functionality.

KEYWORDS

ESP8266, ESP-12F, nodemcu, IoT, Wi-Fi, LoRa, Z-wave, Sigfox, ZigBee, HTTP, CoAP, MQTT, control module

HAMAN, Martin. *Kontrolní a řídicí modul s IoT*. Brno, 2018, 55 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Zoltán Szabó, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Kontrolní a řídicí modul s IoT“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zoltánu Szabó, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Pojem internet věcí – Internet of Things (zkráceně IoT) poprvé použil v roce 1999 ve své stejnojmenné prezentaci britský technologický průkopník Kevin Ashton. Jedná se o označení věcí (především elektrospotřebičů a senzorů), které jsou nějakým způsobem připojeny k internetu. Připojení k internetu je využíváno ke sdílení získaných dat a k dálkovému řízení těchto věcí. Data mohou být sdílena i mezi jednotlivými IoT zařízeními. Získávání a sdílení těchto dat má vést k jejich lepšímu porozumění a následně k ekonomickým úsporám, zjednodušení a optimalizaci lidské činnosti.

Tato bakalářská práce se zabývá problematikou přenosu dat z řídících modulů IoT a jeho návrhem. Při návrhu řídícího modulu IoT je velice důležité správně zvolit způsob přenosu dat. Ten se především odvíjí od velikosti, četnosti přenášených dat, způsobu jejich zpracování a v neposlední řadě taky od způsobu napájení řídícího modulu. Z těchto důvodů je v prvních dvou kapitolách popsáno a porovnáno několik vybraných bezdrátových technologií a aplikačních protokolů. Následující kapitola se věnuje samotnému návrhu řídícího modulu IoT pro vzdálené řízení vytápění. V dalších kapitolách je popsána problematika a složení vytvořeného firmwaru pro řízení vytápění, způsoby zobrazení aktuálních hodnot a ovládání navrženého řídícího modulu IoT, možnosti realizace MQTT broakeru. Vzhledem k tomu, že tento modul bude provozován v domácnosti, byla vybrána bezdrátová technologie WiFi, která je již ve většině domácností běžně používána a není zapotřebí dalších investic.

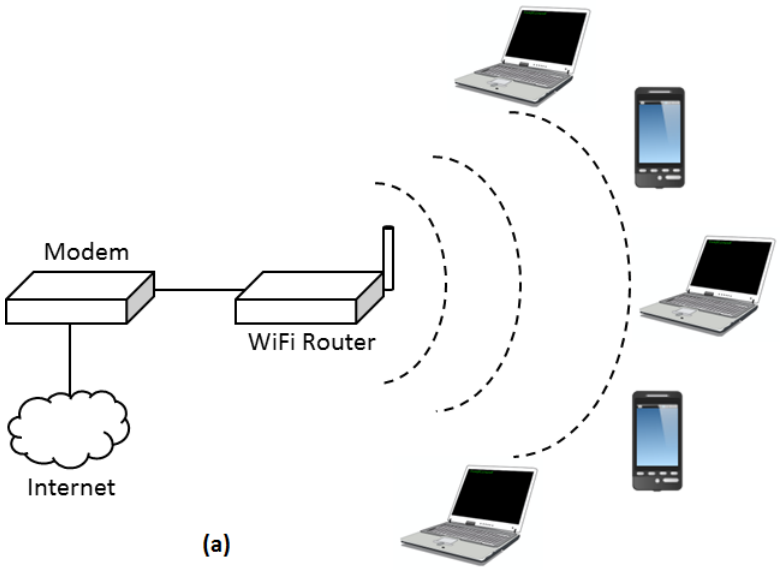
1 MOŽNOSTI BEZDRÁTOVÉHO SBĚRU DAT ZE VZDÁLENÝCH PŘÍSTROJŮ

Tato kapitola se věnuje popisu několika významných bezdrátových technologií, které jsou používány IoT zařízeními.

1.1 WiFi

WiFi byla vytvořena za účelem bezdrátového připojení mobilních zařízení do počítačové sítě ethernet. Postupem času začala sloužit pro bezdrátové připojení k internetu.

Komunikace probíhá v bezlicenčních frekvenčních pásmech 2,4 GHz a 5 GHz, je definována standarty IEEE 802.11. Jednotlivá zařízení se většinou připojují k AP (tvz. přístupový bod – Access Point) a tvoří tak architekturu sítě typu hvězda (viz obr. 1.1). Pro identifikaci má každé AP vlastní SSID (identifikátor bezdrátové sítě WiFi – Service Set Identifier), který v pravidelných intervalech vysílá. Jelikož se využívá sdílené médium, používá se zde metoda CSMA/CA. Tato metoda spočívá v tom, že jednotlivá zařízení připojená k AP naslouchají, zda je médium volné, a v případě, že neprobíhá žádná komunikace, zahájí vlastní komunikaci. Jelikož WiFi zajišťuje pouze komunikaci na spojové vrstvě, je nutné pro přenos informace využít vyšších protokolů k sestavení ethernetového rámce.



Obr. 1.1: Typická architektura WiFi sítě [3].

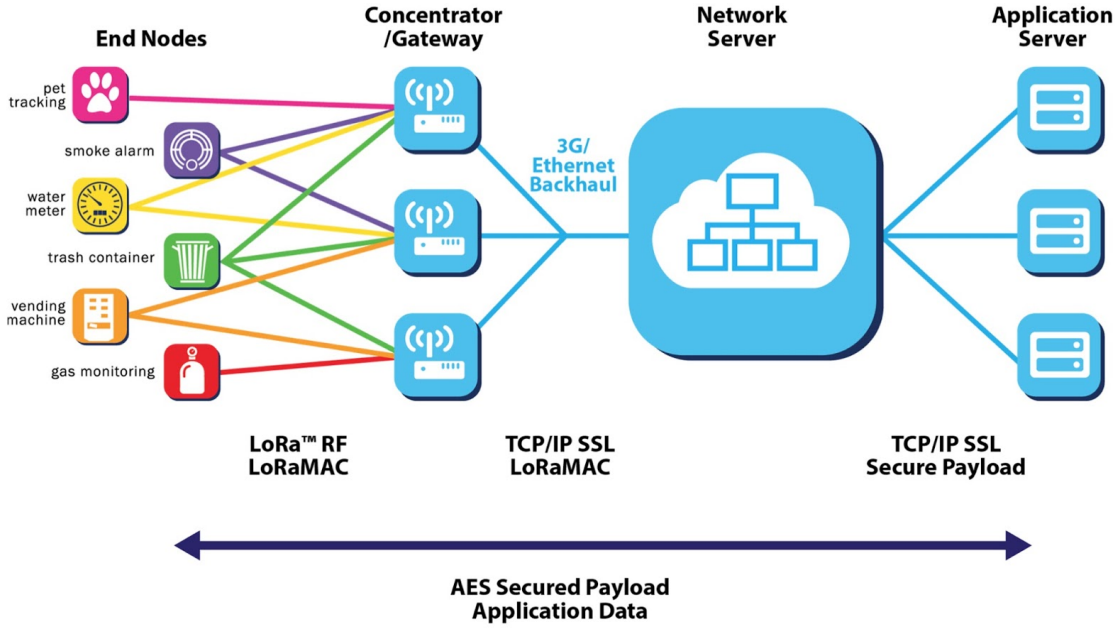
Zabezpečení WiFi lze realizovat:

- **Skrytím SSID**
  AP přestane pravidelně vysílat svoje SSID. Zařízení, které se chce připojit, musí znát SSID tohoto AP. Tento způsob zabezpečení je ale velice neefektivní. SSID není šifrované a je ho možné odposlechnout při komunikaci mezi připojeným zařízením s AP. Lze jej ovšem kombinovat se všemi metodami zabezpečení WiFi.
- **Filtrováním MAC adres**
  AP má seznam povolených zařízení a jejich MAC adres. Při připojování zařízení AP porovná MAC adresu zařízení se seznamem povolených zařízení, a pokud zjistí, že je zařízení v seznamu, tak jej připojí. Útočník odposlouchávaním komunikace mezi AP a připojeným zařízením může MAC odposlechnout a duplikovat ji. Pokud bude připojeno do sítě více zařízení se stejnou MAC, s největší pravděpodobností nastanou problémy v síti. Toto zabezpečení lze kombinovat se všemi metodami zabezpečení WiFi.
- **WEP**
  využívá k zabezpečení symetrické šifry a statické klíče WEP, které se ručně nastavují na obou stranách. Útočník může zachytit mezi AP a připojeným zařízením pakety, ve kterých se klíč přenáší a následně z nich dekódovat klíč. Z těchto důvodů se v dnešní době od tohoto šifrování komunikace odstupuje.
- **WPA**
  je zdokonalení WEP šifrování, kdy se pomocí určitého algoritmu dynamicky mění WEP klíč.
- **WPA2**
  je ze všech dříve zmíněných zabezpečení nejbezpečnější. K šifrování komunikace využívá šifry AES (standard pokročilého šifrování – Advanced Encryption Standard). Nevýhodou jsou větší požadavky na výpočetní výkon jak AP, tak i připojeného zařízení.

Dosah WiFi se pohybuje v rozsahu od desítek metrů po několik kilometrů (v závislosti na prostředí, vysílacím výkonu a použité anténě). Umožňuje přenos dat rychlostí až 600 Mbps. WiFi je díky své pořizovací ceně, jednoduchosti aplikace, více účelovosti velice rozšířená technologie. Při využití této technologie pro IoT (internet věcí – Internet of Things) je výhodou možnost využití více aplikačních protokolů pro přenos dat, přímý přenos dat na server a často je možné využít již existující WiFi síť. Nevýhodou je větší energetická náročnost.

1.2 LoRa

LoRa je standart fyzické vrstvy pro rádiovou komunikaci využívající bezlicenčních pásem 433 MHz, 868 MHz, 915 MHz. Díky využití modulační metody „rozprostření spektra“ vyniká velkým dosahem (až 40 km) a malou spotřebou energie. Jednotlivá zařízení se připojují na brány, které většinou zprostředkovávají komunikaci přes počítačovou síť ethernet a aplikační servery na ní (viz obr. 1.2). Architektura sítě je typu částečné meshe [4]. Svými vlastnostmi splňuje kritéria LPWAN (Nízkoenergetická globální síť – Low Power Wide Area Network).



Obr. 1.2: Architektura sítě LoRa [7].

Pro komunikaci se využívá komunikační protokol LoRaWAN. Ten rozděluje komunikaci na dvě části - aplikační (přenos jednotlivých zpráv ze zařízení) a síťovou (zabezpečení spojení). Obě tyto části jsou šifrované pomocí 128-bit AES [5].

Dále rozeznává tři třídy zařízení:

- **Třída A**
  je zaměřena na maximální úsporu energie. Poté, co zařízení vyšle rámec, umožňuje po dobu dvou rámců příjem.
- **Třída B**
  je zaměřena na menší zpoždění. Vychází ze třídy A, ale v pravidelných intervalech zařízení umožňuje příjem ping rámců.
- **Třída C**
  je zaměřena na co nejmenší zpoždění. Pokud zařízení nevysílá, umožňuje neustále příjem.

LoRa umožňuje přenos dat rychlostí až 50 kbps[6]. Tato technologie je vhodná především pro zařízení, jejichž aplikace vyžadují přenos malého množství dat na větší vzdálenosti a úsporu energie (např. senzor napájený baterií).

1.3 Z-wave

Z-wave je bezdrátová technologie vytvořená za účelem domácí automatizace. Pro komunikaci využívá bezlicenční pásmo 868 MHz.

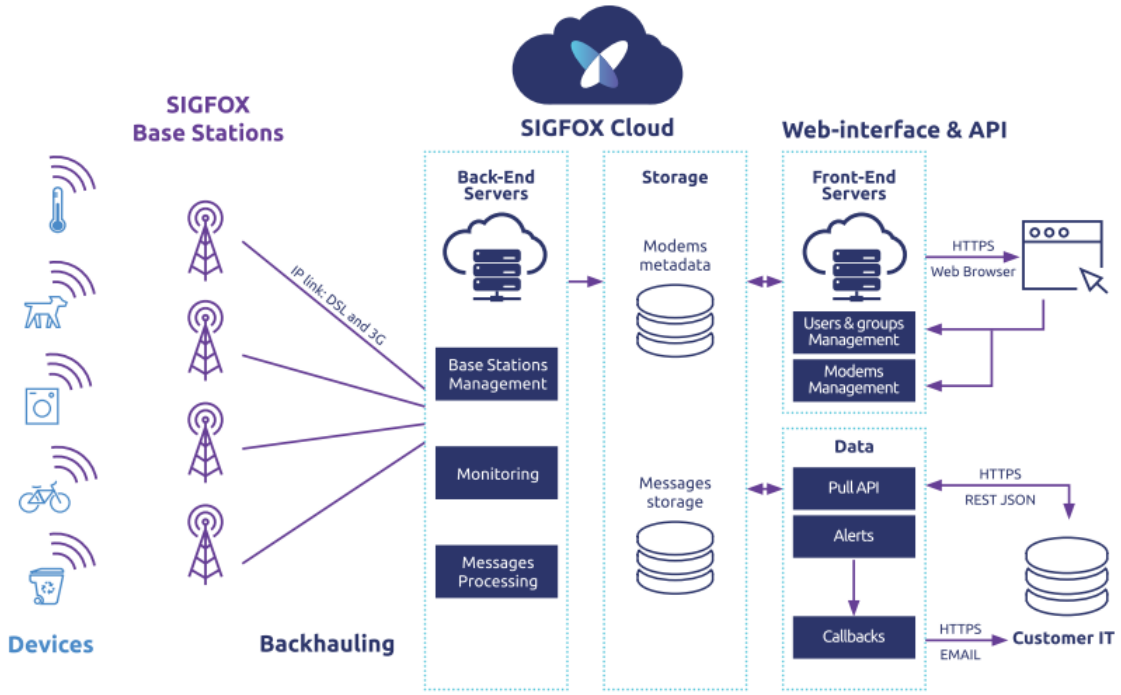
Základním bodem této sítě je tvz. controller (často je integrovaný ve WiFi routeru), který síť vytváří a definuje network ID (4 bytové identifikační číslo sítě). Do této sítě se připojují jednotlivá zařízení, která mají svůj vlastní identifikátor node ID (1 bytové číslo). Do sítě je vzhledem k délce node ID možné přidat pouze 232 zařízení. Pro přidání koncového zařízení se na něm stiskne kombinace tlačítek a poté ho může controller vyhledat. Controller si následně vytvoří záznam ve své směrovací tabulce s ID zařízením. Zařízení si uloží ID sítě. Jelikož z-wave má architekturu typu mesh (kromě koncových zařízení napájených z baterií může každé zařízení přeposílat data dalším zařízením). Může nastat situace, kdy koncové zařízení, které chceme přidat, není v dosahu controlleru. V tomto případě je nutné umístit zařízení na místo, kde bude provozováno, a dočasně přemístit controller po dobu přidávání zařízení do sítě. Z těchto důvodů bývají controllery vybaveny baterií.

Díky vývoji koncových zařízení velkovýrobci elektrotechniky nabývá tato technologie na popularitě. Vzhledem k tomu, že se jedná o komerční technologii, je problematické vyvíjet vlastní koncová zařízení, protože dokumentace je neveřejná.

1.4 Sigfox

Sigfox je další technologií využívající bezlicenční pásmo 868 MHz. Byl navržen pro potřeby IoT a splňuje požadavky pro sítě LPWAN.

Sigfox využívá šířku pásma 192 kHz, ve kterém se přenášejí zprávy o šířce pásma 100 Hz. Pro zajištění kvality přenosu dat IoT zařízení vyšle postupně tři stejné rámce s různým časovým odstupem a na různých frekvencích. Zprávy od IoT zařízení přijímají buňky sigfox operátora, které jsou v dosahu (viz obr. 1.3). Zprávy od IoT zařízení mohou být zároveň přijaty na několika buňkách. Po 20 s od odeslání prvního rámce umožňuje IoT zařízení příjem zprávy po dobu 25 s. Komunikace mezi sigfox operátorem a zákazníkem probíhá přes internet různými způsoby.



Obr. 1.3: Architektura sítě Sigfox [9].

Každá zpráva, obsahuje sekvenční číslo a validační okno. Sekvenční číslo umožňuje odstranit duplicitu zpráv, které jsou ukládány do databáze sigfox operátora. Validační okno slouží k validaci zprávy. Jedná se o symetrickou šifru využívající unikátního klíče, který je zadán do integrovaného obvodu, zprostředkovávajícího komunikaci pomocí sigfox sítě při výrobě. Zprávy ve výchozím nastavení nejsou šifrovány. Je možné uživatelské end to end šifrování nebo je možné využít šifrování, které nabízí Sigfox protokol. Toto šifrování bylo navrženo s ohledem na malou délku zpráv ve spolupráci se společností CEA-LETI. Vychází z unikátního klíče zařízení

a sekvenčního čísla. Vzhledem k regulaci používání bezlicenčního pásma v evropských státech, smí IoT zařízení odeslat 140 zpráv o délce 12 bytů a přijmout 4 zprávy o délce 8 bytů za den.

Tato technologie je vhodná pro IoT zařízení, která jsou vzdálená nebo se pohybují a jejichž aplikace vystačí s malým přenosem dat (např. GPS lokátor vozidla proti odcizení). Další nevýhodou je zpoplatnění sigfox operátorem.

1.5 ZigBee

ZigBee využívá taktéž bezlicenčních pásem 868 MHz, 902–928 MHz a 2,4 GHz. Bylo vytvořeno pro průmyslové a lékařské účely, spotřební elektrotechniku, počítačové periférie a automatizaci budov. Při návrhu bylo myšleno i na možnost implementace ZigBee protokolu do mikroprocesorů s omezenými hardwarovými možnostmi.

Je definováno standardem IEEE 802.15.4 . Přenášená data jsou modulována pomocí QPSK (digitální modulace založená na kvadraturním klíčování fázovým posuvem – Quadrature phase-shift keying) a DSSS (technika přímého rozprostřeného spektra – Direct Sequence Spread Spectrum). Pro přístup ke sdílenému médiu se využívá podobně jako u WiFi metoda CSMA/CA. Komunikace je na síťové vrstvě šifrována pomocí šifrovací metody AES s klíčem o délce 128 bitů.

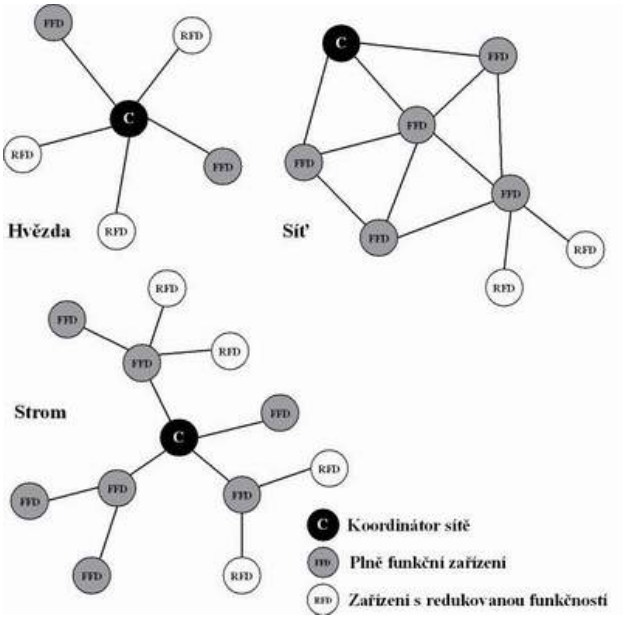
Jelikož různá zařízení mají různé požadavky na přenos dat, jsou definovány tyto režimy přenosu:

- **periodicky se opakující přenosy**
  např. přenos naměřených hodnot ze senzorů.
- **nepravidelné přenosy**
  např. přenos dat při stisku tlačítka.
- **periodické přenosy s požadavkem na malou latenci**
  např. přenos dat z počítačových periférií.

Dále jsou specifikovány typy přenášených rámců:

- **Data Frame**
  je rámec určený pro přenos dat.
- **Acknowledgement Frame**
  je rámec sloužící pro potvrzování přijetí rámce na linkové vrstvě.
- **Beacon Frame**
  je rámec používaný koordinátorem sítě pro přepnutí zařízení do úsporného módu. Zařízení se po určité době opět aktivuje a je schopno příjmu a odesílání dat.
- **MAC Command Frame**
  slouží k řízení a nastavení zařízení a jejich připojení do sítě.

Fyzická vrstva ZigBee umožňuje tři topologie sítě - hvězda, strom, mesh (viz Obr. 1.4). Hlavním bodem sítě je koordinátor, který síť zakládá, definuje její PAN ID (2 bytový identifikátor sítě) a spravuje ji. Zařízení se dle implementace ZigBee protokolu dělí na FFD (zařízení s plnou funkčností – Full Functional Device) a RFD (zařízení s omezenou funkčností – Reduced Functionality Device). RFD se používá na koncových zařízeních, kde je vyžadována co nejmenší hardwarová náročnost. Jednotlivá zařízení jsou adresovaná pomocí 8 bytové nebo zkrácené 2 bytové adresy.



Obr. 1.4: Architektura sítě ZigBee [11].

Aplikační vrstva ZigBee se skládá z těchto částí:

- **Pomocné aplikační APS podvrstvy**
  pomocí párovací tabulky zajišťuje párování zařízení podle poskytovaných služeb a požadavku.
- **ZigBee objektů (ZDO)**
  ZigBee objekt definuje typ zařízení (koordinátor, směrovač, koncové zařízení), umožňuje vyhledávání jednotlivých zařízení a zajišťuje nastavení zabezpečení.
- **Uživatelských aplikačních objektů**
  definuje typ koncového zařízení a formát zpráv. Uživatelské aplikační profily jsou označeny 2 bytovým identifikátorem.

Dosah ZigBee se pohybuje kolem 50 m a umožňuje přenos dat rychlostí až 250 kbps. Je vhodné pro aplikace vyžadující spolehlivý přenos menšího množství dat na kratší vzdálenosti.

2 APLIKAČNÍ PROTOKOLY PRO PŘENOS DAT Z IOT ZAŘÍZENÍ

V této kapitole je popsáno několik vybraných aplikačních protokolů použitelných pro přenos dat z řídících modulů IoT. Výběr aplikačního protokolu se odvíjí od hardwarových možností řídícího modulu IoT a od způsobů, jak s daty bude nakládáno.

2.1 HTTP protokol

Protokol HTTP (Hypertext Transfer Protocol) byl vytvořen za účelem přenosu HTML dokumentů. Přenos probíhá pomocí navazovaného spojení TCP a portu 80. Komunikace je založena na tom, že klient generuje požadavky a server generuje na tyto požadavky odpovědi. K tomu se využívají následující typy zpráv: GET - požadavek klienta na zaslání stránky, PUT - nahrání souboru na server, POST - nahrání uživatelských dat na server (např. formuláře).

Výhodou tohoto protokolu je jednoduchý přístup k datům skrze webový prohlížeč, který se v dnešní době nachází na většině PC, tabletů a chytrých mobilních zařízení. Nevýhodou je nutnost znát IP adresu daného řídícího modulu IoT, obtížnější předávání dat mezi řídícími moduly IoT, velké paměťové nároky pro uložení HTML dokumentu a hlavně při doplňování dat do HTML dokumentu. Posledně zmíněnou nevýhodu lze částečně odstranit za použití HTML dokumentu obsahujícího JavaScript (skript, realizující se ve webovém prohlížeči klienta), který může navázat s řídícím modulem IoT spojení za použití protokolu WebSocket, přenést tak potřebná data a zobrazit je.

2.2 CoAP protokol

Protokol CoAP (Constrained Application Protocol) vychází z HTTP protokolu. Byl přímo navržen pro komunikaci s IoT zařízeními. Z těchto důvodů se od HTTP protokolu v několika věcech zásadně liší. Především textová hlavička s velkým množstvím parametrů byla nahrazena jednodušší bitovou hlavičkou s menším počtem parametrů. Data nejsou přenášena pomocí HTML dokumentu, ale pomocí krátkých zpráv. Komunikace probíhá pomocí nenavazovaného spojení UDP.

Těmito změnami se odstranila hlavní nevýhoda HTTP protokolu spočívající ve velkých paměťových nárocích. Nevýhodami tohoto protokolu je nutnost znát IP adresu daného řídícího modulu IoT a pro přístup k datům použití speciální aplikace nebo webových stránek, které získají a zobrazí data z řídícího modulu IoT.

2.3 MQTT protokol

Tento protokol byl vybrán pro komunikaci s navrhovaným řídícím modulem IoT díky své jednoduchosti, malým hardwarovým nárokům a tomu, že data jsou ukládána na jednom místě (brokeru).

2.3.1 Vznik MQTT protokolu

Protokol MQTT (Queuing Telemetry Transport) byl vytvořen v roce 1999 Andy Stanfordem (IBM) a Arlenem Nipperem (Eurotech) pro řízení ropovodů. Brzy se však začal díky své jednoduchosti, efektivnosti a malé hardwarové náročnosti používat v zařízeních pro domácí automatizaci. V roce 2013 byl certifikovaný mezinárodní standardizační společností OASIS jako open source protokol vhodný pro komunikaci mezi jednotlivými zařízeními.

2.3.2 Způsob posílání zpráv

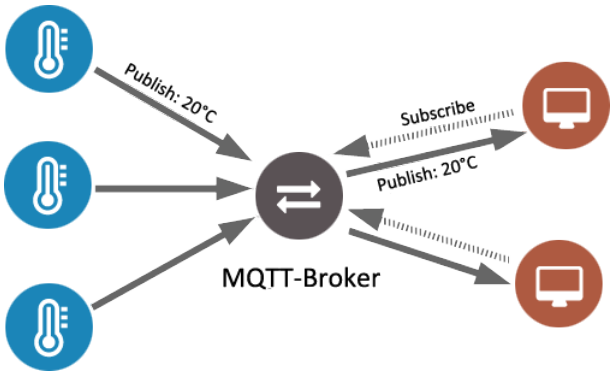
Komunikace probíhá pomocí zpráv s binární hlavičkou a textovým obsahem zprávy. V těchto zprávách se nejčastěji přenáší text, číselné hodnoty a JSON (pole, tabulky nebo jiné struktury formátované do textového řetězce). Rozesílání zpráv je založeno na metodě publish, subscribe (publikuj, odebírej). Hlavním bodem komunikace pomocí MQTT protokolu je broker. Broker je „server“, který se stará o příjem, rozesílání zpráv a případné jejich ukládání. Klient (koncové zařízení) pošle zprávu spadající pod určité téma brokeru. Ten zjistí ze svého seznamu klienty, kteří si zažádali o odběr daného tématu. Těm následně zprávu rozešle. Jednotliví klienti mohou být publikátory a zároveň i odběrateli několika témat.

2.3.3 Připojení klientů k brokeru

Jednotliví klienti se připojují pomocí TCP k brokeru (viz obr. 2.1) nejčastěji přes port 1883. V případě šifrovaného spojení TLS je použit port 8883. Pro připojení klienta může broker vyžadovat autentizaci pomocí přihlašovacího jména a hesla. Při připojování posílá klient zprávu CONNECT většinou i s příznakem „clean session“, který zajistí odregistrování odběru všech témat, která mohla být dříve přednastavena. Broker při úspěšném připojení klienta potvrdí jeho připojeni pomocí zprávy CONACK.

Po úspěšném připojení k brokeru si může klient zaregistrovat odběr některých témat pomocí zprávy SUBSCRIBE. Úspěšné nastavení odběru témat potvrdí broker klientovi pomocí zprávy SUBACK. Stejně tak si může klient pomocí zprávy UNSUBSCRIBE odregistrovat odběr témat. Broker v tomto případě posílá klientovi taktéž potvrzující zprávu UNSUBACK.

Aby bylo možné zjistit, jestli je klient stále aktivní, v případě, že negeneruje žádné zprávy, posílá v pravidelných intervalech zprávy PINGREQ, které broker potvrdí pomocí zprávy PINGACK.



Obr. 2.1: Architektura komunikace pomocí MQTT protokolu [16]

2.3.4 Dělení zpráv do témat

Jednotlivé zprávy jsou dělené do témat. Každá zpráva má své vlastní téma. Téma může obsahovat další podtémata nebo jednu zprávu. Jsou hierarchicky uspořádaná, oddělena pomocí „/“, podobně jako je tomu například v zápisu cesty k souboru na PC. Témata jsou v MQTT protokolu zapsána pomocí řetězce ve formátu UTF-8, takže je možné v názvech témat používat diakritiku.

Struktura témat není fixně definovaná. Každý návrhář klientů, využívající MQTT protokol, si může navrhnout vlastní strukturu témat. Není ji nutné předem definovat, broker ji vytváří na základě příchozích zpráv a jejich zařazení do témat.

Aby při odběru všech podtémat a zpráv spadajících do určitého nadřazeného tématu jsme nemuseli každé podtéma se zprávou jednotlivě definovat, je možné použít znak „#“ (například „*budova-1/podlaží-2/#*“). Budeme-li chtít odebírat určité zprávy, u kterých se liší jedna úroveň nadřazených témat, je ji možno nahradit znakem „+“ (například „*budova-1/podlaží-2/+/teplota*“, kde znak „+“ nahrazuje jednu úroveň témat označující jednotlivé místnosti na daném podlaží).

2.3.5 QoS

QoS (kvalita služeb (přenosu) – Quality of Service) určuje, jakým způsobem má být zpráva doručena a jak má být zajištěna spolehlivost doručení zprávy. Různé typy zpráv vyžadují rozdílné zajištění spolehlivosti doručení. Například bude-li se odesílat každou sekundu zpráva o teplotě v místnosti, většinou nebude vadit, pokud nějaká zpráva nebude doručena, protože ji po chvíli nahradí nová zpráva. Pokud by bylo potvrzováno každé přijetí zprávy, případně opakování přenosu takového charakteru zpráv, vedlo by to k zbytečnému zatížení sítě a ve výsledku by to bylo kontraproduktivní. Proto MQTT protokol rozeznává tři úrovně QoS:

- **Úroveň 0: at most once**
  publisher vyšle zprávu PUBLISH brokeru a neočekává potvrzení přijetí zprávy. Broker rozešle všem odběratelům tuto zprávu bez ověření doručení.
- **Úroveň 1: at least once**
  publisher vyšle zprávu PUBLISH brokeru. Broker rozešle všem odběratelům tuto zprávu a čeká na potvrzení přijetí PUBACK od odběratelů. Jakmile dostane od všech odběratelů potvrzení PUBACK, odešle také potvrzení přijetí zprávy PUBACK publisherovi.
- **Úroveň 2: exactly once**
  Publisher odešle zprávu brokeru a ten ji rozešle odběratelům. Po té broker odešle potvrzení přijetí PUBREC, publisher potvrdí přijetí potvrzení pomocí zprávy PUBREL. Následně broker ukončí komunikaci komunikaci pomocí zprávy PUBCOMP.

Broker rozesílá odběratelům zprávy se stejnou úrovní QoS, s jakou je přijal. Pokud odběratel zprávy nepodporuje danou úroveň QoS, dojde k její změně úrovně na takovou, kterou podporuje. Dále je možné nastavit pomocí příznaku retain, jestli má broker zprávu po rozeslání odběratelům smazat nebo uchovat, např. pro nové odběratele.

2.3.6 Výhody použití MQTT protokolu

Hlavní výhodou MQTT protokolu je, že pro komunikaci a získávání dat není nutné znát IP adresu jednotlivých klientů, ale pouze IP adresu brokeru. Veškerá aktuální data jsou uložena na jednom místě. Předávání zpráv mezi jednotlivými klienty je možné řešit centrálně pomocí skriptu, který se připojí k brokeru jako klient. Což je uživatelsky přívětivější, než definovat každému klientovi, jaké má odebírat zprávy od jiných klientů a jak s nimi má nakládat. Obzvláště, dojde-li ke změně klientů. Pomocí obdobných skriptů a aplikací je možné řešit i ukládání zpráv do databáze, jejich vyhodnocování, zobrazení a řízení, např. pomocí webového rozhraní.

3 NÁVRH ŘÍDÍCÍHO MODULU IOT

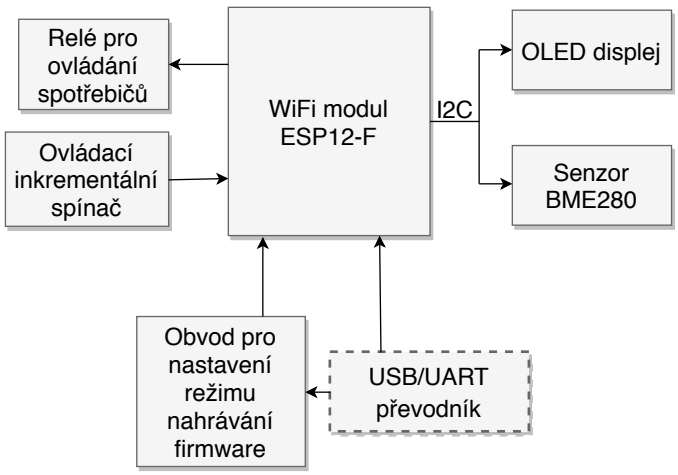
Tato kapitola se zabývá návrhem řídícího modulu IoT a popisem jednotlivých částí.

3.1 Požadované funkce a vlastnosti řídícího modulu

Navrhovaný řídící modul IoT bude připojen k počítačové síti pomocí bezdrátové technologie WiFi. Komunikace s tímto modulem bude probíhat za použití vybraného MQTT protokolu. Modul bude schopný měřit teplotu, tlak a vlhkost okolního vzduchu. Naměřené hodnoty budou zobrazeny na OLED displeji. Na základě nastavení a naměřených hodnot bude schopný ovládat spotřebiče pomocí relé. Pro případ výpadku napájení bude záložně napájen pomocí Li-Ion baterie, což bude schopný detekovat.

3.2 Mikroprocesorová platforma a periférie

Hlavním bodem celého řídícího modulu je WiFi modul ESP-12F (viz obr. 3.1). K němu je pomocí jedné I2C sběrnice připojen OLED displej pro zobrazení naměřených hodnot a senzor Bosch BME280 pro měření teploty, tlaku a vlhkosti ovzduší. Dále je k WiFi modulu připojen obvod s relé pro ovládání spotřebičů a inkrementální spínač, který může být použit pro ovládání uživatelského prostředí při případném dalším vývoji.

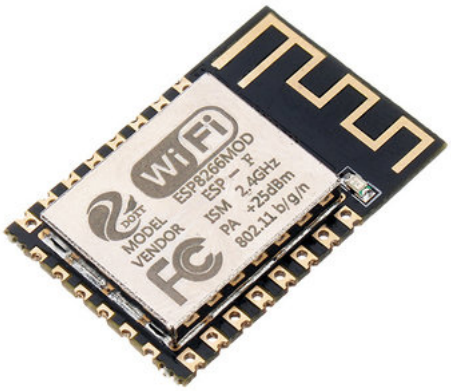


Obr. 3.1: Blokové schéma mikroprocesorové platformy a periférií.

Firmware bude nahráván do WiFi modulu ESP-12F pomocí externího USB/UART převodníku. K tomu, aby bylo možné nahrát do WiFi modulu, je nutné nastavit nízkou úroveň na pinu GPIO 0 a modul resetovat krátkodobým přivedením nízké úrovně na pin RST. O toto nastavení se stará obvod pro nastavení režimu nahrávání firmware, který toto nastavení provede na základě signálů DTR a RTS z USB/UART převodníku.

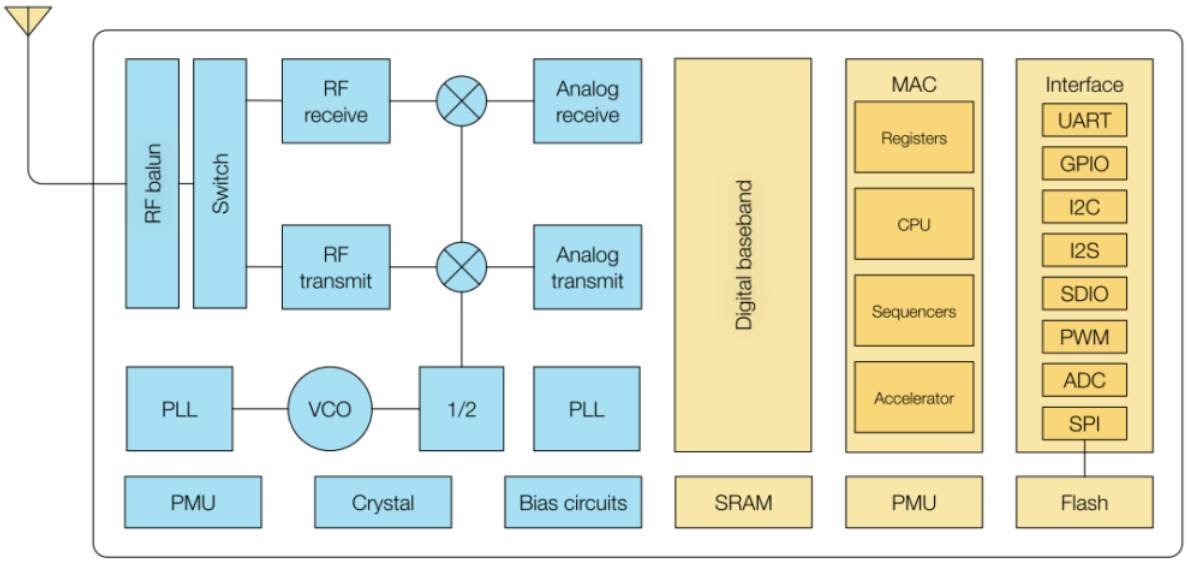
3.2.1 Mikroprocesorová platforma

Jako mikroprocesorová platforma byl vybrán WiFi modul ESP-12F (viz obr. 3.2), který vyniká svoji cenou a výpočetním výkonem. Je založený na obvodu Espressif ESP8266EX. Kromě tohoto obvodu obsahuje ještě 4 MB ISP flash paměť a 3 dBi anténu. Výhodou tohoto modulu oproti ostatním modulům založeným na obvodu ESP8266EX je, že má vyvedené všechny GPIO piny. K pinům GPIO 9 až 14 je připojena vnitřní flash paměť a není je možné dále využít. Modul ESP-12F se od známějšího modulu ESP-12E liší hlavně v lepší konstrukci antény.



Obr. 3.2: Modul ESP-12F [18].

Obvod Espressif ESP8266EX je vysoce integrovaný SoC (integrovaný obvod zahrnující všechny potřebné součásti – System on Chip) WiFi obvod (viz obr. 3.3), založený na 32-bitovém mikroprocesoru Tensilica L106. ESP8266EX již zajišťuje veškerou komunikaci skrze WiFi (připojení, autentizace, přenos dat, šifrování,...), na což je využito 20 % výpočetního výkonu tohoto obvodu.



Obr. 3.3: Blokový diagram obvodu ESP8266EX [20].

Parametry integrovaného obvodu ESP8266EX:
rozsah vstupního napětí: 3 - 3,6 V,
maximální odběr proudu: 170 mA,
frekvence MCU: 80/160 MHz,
podporované WiFi protokoly: 802.11 b/g/n,
WiFi režimy: station, AP, AP+station,
WiFi zabezpečení: WPA, WPA2,
šifrování: WEP, TKIP, AES,
sběrnice: I2C, SPI, UART,
funkce: PWM, ADC (10 bit).

3.2.2 OLED display

Pro možnost zobrazení naměřených hodnot teploty, tlaku, vlhkosti vzduchu a případné nastavení byl vybrán grafický OLED display. Tento displej byl zvolen z důvodu nízkého napájecího napětí (3,3 V) a vzhledem k malému počtu GPIO pinů modulu ESP-12F kvůli možnosti komunikovat pomocí I2C sběrnice, která vyžaduje pouze 2 tyto piny. Jelikož vývod tohoto displeje je realizován pomocí flex kabelu, je pro snazší použití již připájen k plošnému spoji spolu se součástkami nutnými pro provoz tohoto displeje (viz obr. 3.4).

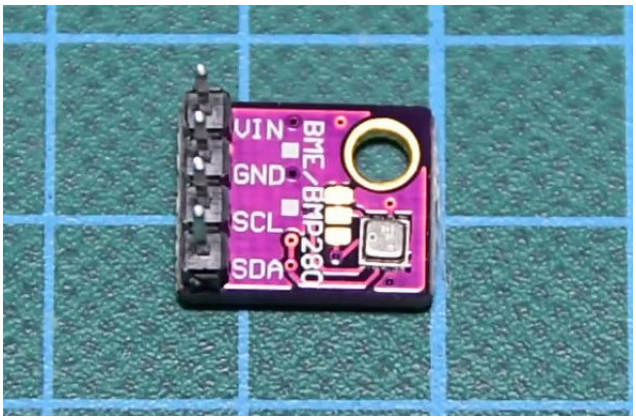
Parametry OLED displeje:
napájecí napětí: 3,3 - 5 V,
sběrnice: I2C, SPI - není vyvedeno na plošném spoji,
řadič: SSD1306,
rozlišení: 128 x 63 pixelů,
velikost displeje: 0,96 ", 23 mm x 35 mm.

Obr. 3.4: OLED displej [21].

3.2.3 Senzor BME280

Pro měření teploty, tlaku a vlhkosti ovzduší byl vybrán senzor Bosch BME280. Tento senzor měří všechny požadované veličiny. Pro komunikaci s tímto senzorem lze použít sériové sběrnice I2C. Jelikož I2C umožňuje paralelní připojení několika slave zařízení, lze tento senzor připojit k již dříve zmíněnému OLED displeji a využít tak již použitých GPIO pinů na modulu ESP-12F. Vzhledem k obtížně pájitelnému pouzdru tohoto senzoru je použit modul (viz obr. 3.5), na kterém je již připájený. Kromě senzoru obsahuje tento modul napěťový stabilizátor a obvod pro převod napěťových úrovní, což umožňuje použití 5 V napájení a logiky.

Parametry sensoru Bosch BME280:
napájecí napětí: 3,3 - 5 V,
průměrný odběr proudu: 3,6 μA - při měření teploty, tlaku, vlhkosti s frekvencí 1 Hz,
sběrnice: I2C, SPI - není na modulu vyvedeno,
rozsah měření teploty: -40 - +85 °C,
rozsah měření tlaku: 300 - 1100 hPa,
rozsah měření vlhkosti: 0 - 100 %.

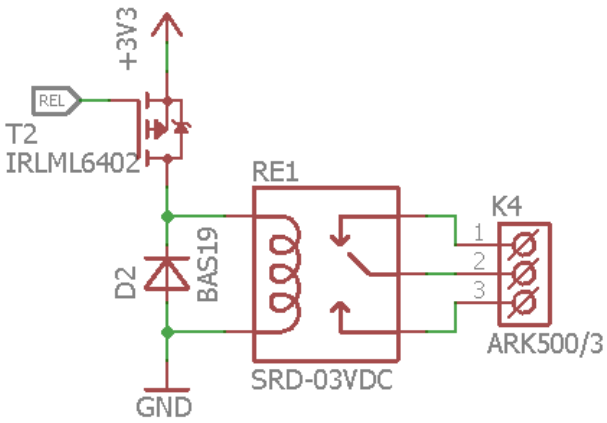


Obr. 3.5: Modul se senzorem Bosch BME280 [22].

3.2.4 Řízení spotřebičů

Pro možnost řízení největší škály spotřebičů je použito relé s výstupy NC (běžně sepnuto – Normally Close) a NO (běžně sepnuto – Normally Close). Jelikož výstupní pin modulu ESP-12F může být zatížen proudem maximálně 12 mA, je nutné relé spínat pomocí tranzistoru (viz obr. 3.6). Pro tento účel byl vybrán již v zapojení použitý unipolární MOSFET tranzistor s P kanálem. Ačkoliv v tomto tranzistoru

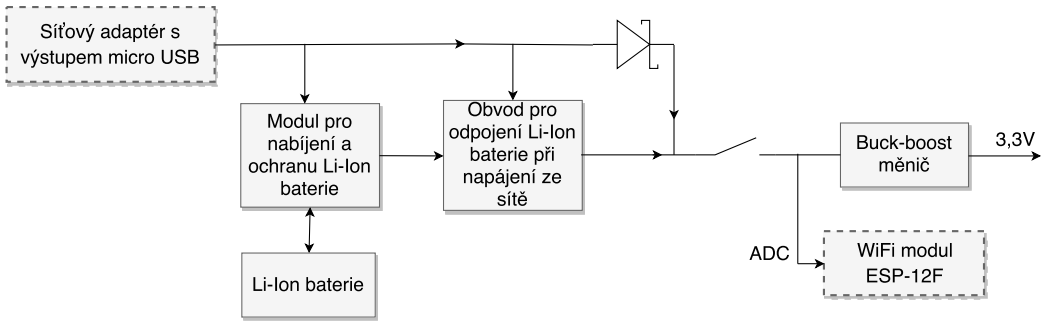
je již obsažena ochranná dioda, byla přidána ochranná dioda D2, aby zbytečně nedocházelo k rušení napájecího napětí 3,3 V.



Obr. 3.6: Zapojení relé pro řízení spotřebičů.

3.3 Napájení

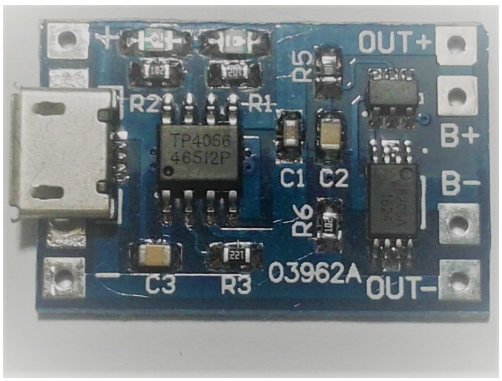
Řídící modul IoT je primárně napájen pomocí síťového adaptéru. Při výpadku síťového napájení je napájen ze záložní Li-Ion baterie. Napájení ze síťového adapteru s výstupem micro USB je přivedeno na modul pro nabíjení a ochranu Li-Ion baterie. Aby nedocházelo k ovlivňování nabíjecího cyklu tohoto modulu, není možné zbytek obvodu neustále napájet přes baterii. Přepínání mezi napájení pomocí síťového adaptéru a baterii je zajištěno obvodem pro odpojení Li-Ion baterie při napájení ze sítě. Následně je pomocí buck-boost měniče dosaženo požadovaného napětí 3,3 V. Detekce výpadku a stavu nabití baterie je zajištěna pomocí snímání velikosti napětí před buck-boost měničem pomocí ADC převodníku WiFi modulu ESP-12F.



Obr. 3.7: Blokové schéma napájecího obvodu.

3.3.1 Modul pro nabíjení a ochranu Li-Ion baterie

Pro záložní provoz řídícího modulu IoT při výpadku napájecího napětí slouží Li-Ion baterie. Pro její nabíjení, ochranu proti podvybití a přebití slouží modul založený na nabíjecím obvodu TP4056 (viz obr. 3.8). Tento modul má micro USB konektor, přes který bude napájen řídicí modul IoT.



Obr. 3.8: Modul pro nabíjení a ochranu Li-Ion baterie.

Nabíjecí obvod TP4056, jehož výrobcem je firma NanJing Top Power ASIC Corporation, slouží k nabíjení jednoho článku Li-Ion baterie. Nabíjecí cyklus tohoto obvodu se dá rozdělit do dvou fází. V první fázi, kdy má baterie napětí menší než 4 V, udržuje konstantní proud. Tento se dá nastavit až na 1 A pomocí rezistoru, který je na modulu označený jako R3. Když baterie dosáhne většího napětí, než je 4 V, přejde tento obvod do druhé fáze nabíjení, kdy baterii nabíjí pomocí konstantního napětí 4,2 V. Stav nabíjení je indikován pomocí dvojice LED diod.

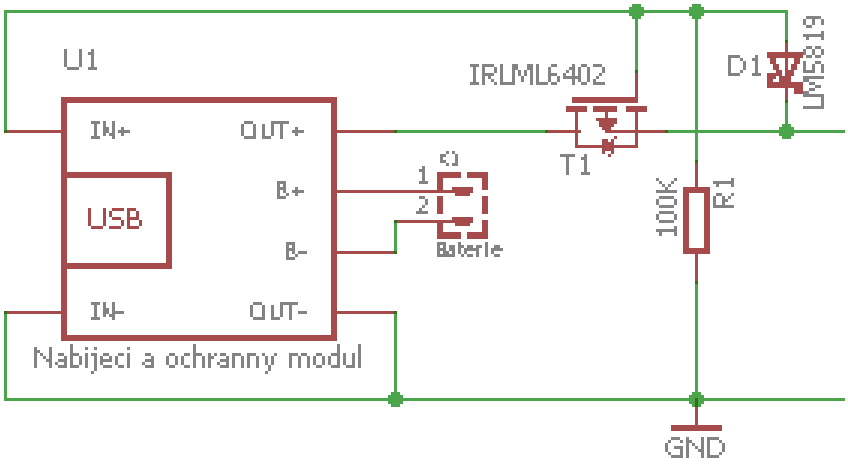
Parametry nabíjecího modulu:
napájecí napětí: 4,5-6 V,
maximální nabíjecí proud: 1 A.

3.3.2 Obvod pro odpojení Li-Ion baterie při napájení ze sítě

Při napájení ze sítě pomocí síťového adaptéru je nutné odpojit zbytek obvodu od Li-Ion baterie, aby nedocházelo k ovlivňování nabíjecího cyklu modulu pro nabíjení a ochranu Li-Ion baterie. K tomuto účelu slouží obvod složený z P-MOSFET tranzistoru T1, Schottkyho diody D1 a rezistoru R1 (viz obr. 3.9).

Je-li řídící modul IoT napájen ze síťového adaptéru, je přiváděno na gate tranzistoru T1 kladné napětí a tranzistor je uzavřen. Zbytek obvodu je napájen přes diodu D1.

V případě záložního napájení je gate uzemněn pomocí R1 a přes vnitřní diodu tranzistoru T1 mezi drain a source začne protékat proud do zbytku obvodu. To způsobí vznik záporného napětí mezi drain a gate. Tranzistor se tím pádem otevře.

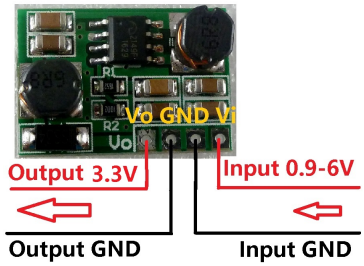


Obr. 3.9: Obvod pro odpojení Li-Ion baterie při napájení ze sítě.

3.3.3 Buck - boost měnič napětí

Celé zařízení je zálohované pomocí jednoho článku Li-Ion baterie, která má v nabitém stavu napětí 4,2 V a ve vybitém stavu má pouze napětí 3,1 V. Zařízení pro svůj provoz vyžaduje napětí 3,3 V. Aby bylo možné využít celou kapacitu baterie, je zapotřebí měniče napětí, který je schopen jak snižovat napětí, tak ho i zvyšovat (tvz. buck and boost měnič). Vzhledem k tomu, že většina řídících integrovaných obvodů pro tyto měniče má takové provedení pouzdra, které je obtížně ručně pájitelné, byl vybrán již hotový modul měniče (viz obr. 3.10).

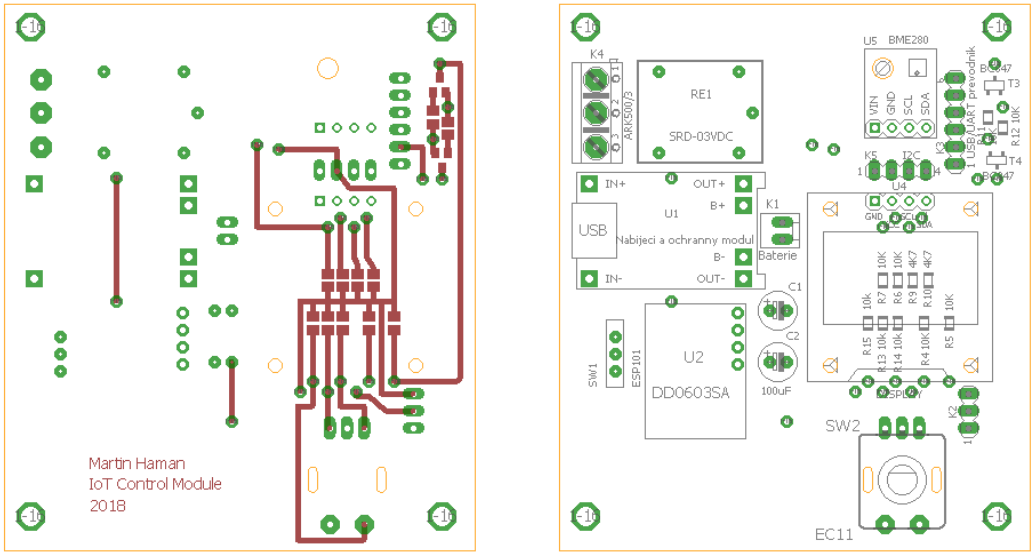
Parametry modulu měniče buck-boost:
rozsah vstupního napětí: 0,9 - 6 V,
výstupní napětí: 3,3 V (±5 %),
maximální výstupní proud: 0,85 A,
pracovní frekvence: 1 MHz,
účinnost: 61 - 85 %.



Obr. 3.10: Modul měniče buck-boost [24].

3.4 Návrh plošného spoje

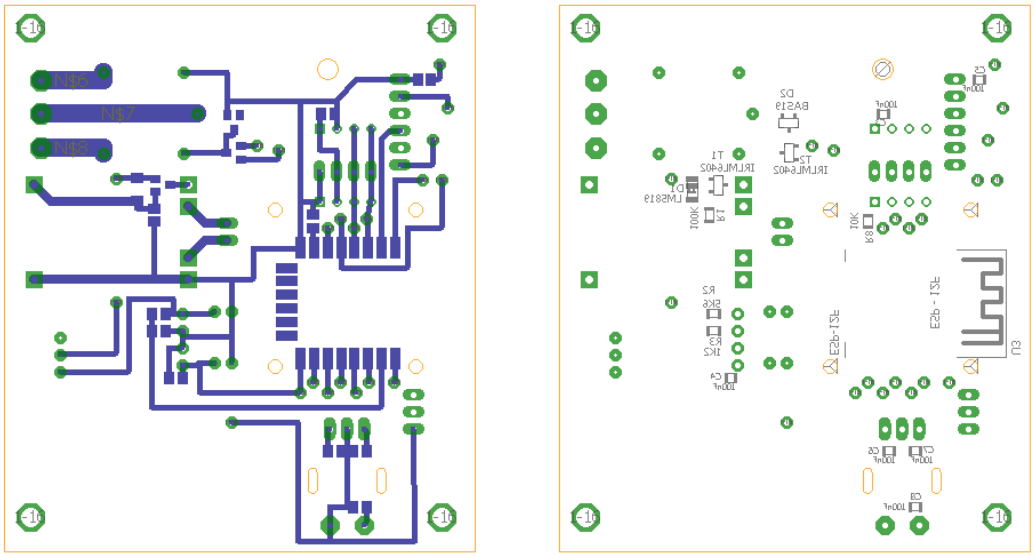
Návrh plošného spoje pro řídící modul IoT byl proveden v CAD systému Eagle od firmy Autodesk. [1] Vzhledem k tomu, že oboustranný plošný (viz obr. 3.11 a obr. 3.12) spoj bude vyroben v domácích podmínkách, jsou prokovy realizovány pomocí vodiče, který je z obou stran plošného spoje připájen. Z těchto důvodů není možné umístění prokovů pod WiFi modul ESP–12F. Oproti původnímu návrhu zapojení řídícího modulu IoT byl na vstup i výstup napěťového měniče buck-boost doplněny kondenzátory pro lepší stabilitu (viz příloha A.1). Tento buck-boost měnič byl v rámci možností umístěn co nejdále od antény WiFi modulu ESP–12F, aby se omezilo rušení přijímaného WiFi signálu. Dále bylo dbáno, aby pod anténou WiFi modulu ESP-12F na plošném spoji procházelo co nejméně spojů. Výsledný navržený dvouvrstvý plošný spoj je přibližně široký 7 cm a dlouhý 8,5 cm.



(a) Návrh. (b) Osazovací plán.

Obr. 3.11: Vrchní strana plošného spoje.

[1] Více informací o programu Eagle jsou dostupné na stránce https://www.autodesk.com/products/eagle/overview



(a) Návrh. (b) Osazovací plán.

Obr. 3.12: Spodní strana plošného spoje.

4 FIRMWARE

Z výroby bývá ve WiFi modulech založených na integrovaném obvodu ESP8266 většinou nahraný firmware pro komunikaci a ovládání modulu pomocí AT příkazů. Výrobce umožňuje vytvoření vlastního firmwaru pomocí poskytnutého SDK (soubor nástrojů pro vývoj softwaru – Software development kit). K dispozici jsou také firmwary, které obsahující interpreter pro skriptovací jazyk Lua a MicroPython nebo podporu Arduino IDE. Výhoda těchto firmwarů je snadnější programování, ladění, dostupnost knihoven pro různé moduly, protokoly apod. Vzhledem k předchozím zkušenostem byl zvolen firmware NodeMCU obsahující interpreter pro skriptovací jazyk Lua.

4.1 NodeMCU

Firmware NodeMCU[1] má v sobě implementovaný Lua interpreter, který umožňuje vykonávat skripty napsané ve skriptovacím jazyce Lua. Tyto skripty je možné nahrávat, kompilovat, odebírat za běhu zařízení. Není tak zapotřebí znovu kompilovat a nahrávat celý firmware. Další výhodou je, že skripty mohou být spuštěny bez kompilace a tak je i jejich ladění snazší. Firmware vychází z SDK výrobce a jeho knihovny jsou napsané v programovacím jazyce C. Knihovny umožňují podporou širokého spektra komponent, protokolu a podobně.

4.2 Kompilace NodeMCU firmwaru

Kompilaci firmwaru lze realizovat dvěma způsoby. Buď pomocí webového kompilátoru [2] nebo stažením zdrojových kódů a jejich vlastním kompilací. Pro kompilaci firmwaru byl vybrán webový kompilátor. Při jeho použití je nutné zadat emailovou adresu a vybrat požadované knihovny, které budou ve firmwaru zkompilovány. Po dokončení kompilace jsou na zadanou emailovou adresu zaslány odkazy pro stažení zkompilovaného firmwaru. Zkompilovaný firmware lze stáhnout ve dvou variantách - inteager (firmware umožňující používat číselné proměnné obsahující pouze celočíselná čísla) a float (firmware umožňující používat číselné proměnné obsahující i desetinná čísla). Vzhledem k tomu, že měřené hodnoty mají desetinnou část, byl použit firmware ve variantě float.

[1] Více informací o firmwaru NodeMCU jsou dostupné na stránce http://nodemcu.readthedocs.io/en/dev/

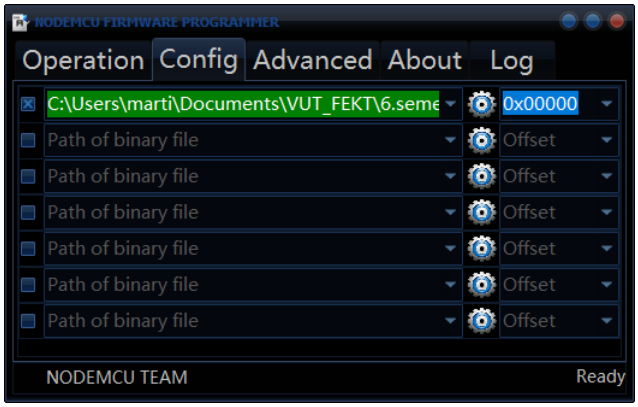
[2] Kompilace NodeMCU firmwaru je možná pomocí stránky https://nodemcu-build.com/

4.3 Nahrání NodeMCU firmwaru

Pro nahrání firmwaru a skriptů do modulu ESP8266 se využívá převodníku USB - UART s logickou úrovni napětí 3,3 V. Aby bylo možné firmware do modulu ESP8266 nahrát, je zapotřebí modul nabootovat do módu UART programování. To se provede pomocí přivedení logické jedničky na pin GPIO 0 a krátkodobým přivedením logické nuly na pin RST. O nabootování do módu UART programování se stará obvod složený z tranzistorů T3 a T4 (viz příloha), který je připojený pomocí konektoru K3 k signálu DTR a RTS zmíněného převodníku.

Dále je zapotřebí software „NODEMCU FIRMWARE PROGRAMMER“ [3] , jehož použití je následující:

1. V záložce „Config“ (viz obr.4.1) se na prvním řádku pomocí ikony „ozubeného kolečka“ vybere již dříve zkompilovaný firmware a vlevo od něj se nastaví offset na hodnotu „0x00000“.



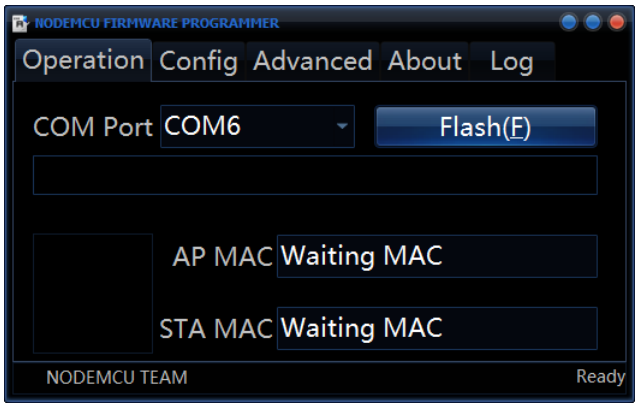
Obr. 4.1: Záložka Config programu NODEMCU FIRMWARE PROGRAMMER.

[3] Program NODEMCU FIRMWARE PROGRAMMER je volně dostupný na stránce https://github.com/nodemcu/nodemcu-flasher

2. V záložce „Advanced“ (viz obr.4.2) je vhodné nastavit „bautrade“ na hodnotu „115200“, „Flash size“ na hodnotu „4MByte“, „Flash speed“ na hodnotu „20MHz“.

Obr. 4.2: Záložka Advanced programu NODEMCU FIRMWARE PROGRAMMER.

3. Nakonec je zapotřebí v záložce „Operation“ (viz obr.4.3) nastavit port použitého převodníku a zmáčknout tlačítko „Flash“, čímž se nahraje se nahraje firmware do modulu.



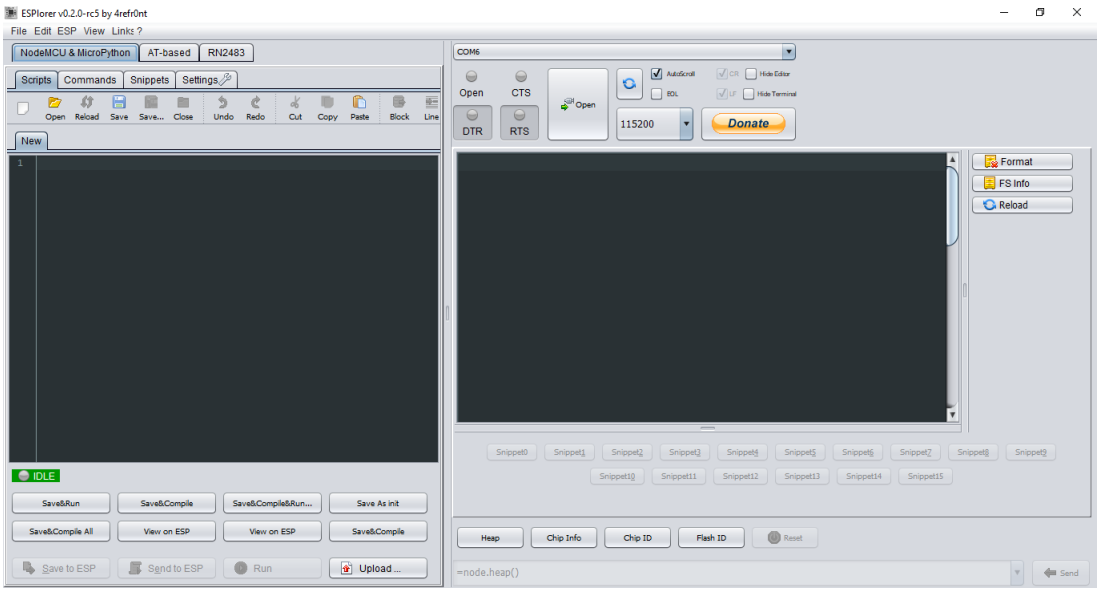
Obr. 4.3: Záložka Operation programu NODEMCU FIRMWARE PROGRAMMER.

4.4 Vytváření a nahrávání Lua skriptů

Pro vytváření skriptů a jejich nahrávání do WiFi modulu ESP8266 slouží volně dostupný program ESPlorer. [4] Tato aplikace se skládá ze dvou hlavních částí - editoru a terminálu (viz obr.4.4).

Levá část okna je vyhrazena pro editor, ve kterém je možné vytvářet a editovat skripty. Editor umožňuje práci s více skripty zároveň, podobně jak je tomu ve webových prohlížečích a zobrazení více stránek v záložkách. Pod editorem se nachází tlačítka pro uložení skriptu, nahrání skriptu do WiFi modulu, kompilaci skriptu a spuštění vykonávání skriptu.

Pravá část okna obsahuje terminál, nad nímž se nachází nastavení sériové linky realizované pomocí převodníku USB - UART s logickou úrovní 3,3 V. Napravo od terminálu se nachází tlačítka pro práci se soubory a skripty uloženými ve WiFi modulu. Po stisku tlačítka "Reload"se pod tímto tlačítkem zobrazí skripty a soubory uložené ve WiFi modulu. Následným najetím kurzorem na soubor a stisku pravého tlačítka myši lze vyvolat kontextovou nabídku, ve které je možné provést kompilaci skriptu, jeho odstranění a nebo stažení z WiFi modulu do počítače. Pod terminálem se nachází tlačítka pro zobrazení informací o WiFi modulu, jeho paměti a zaslání příkazu pro reset WiFi modulu.



Obr. 4.4: Okno programu ESPlorer.

[4] Program ESPlorer je volně dostupný na stránce https://esp8266.ru/esplorer/

4.5 Dělení skriptů na jednotlivé moduly

Z důvodu přehlednosti je vhodné skript rozdělit na jednotlivé části - moduly dle jejich zaměření. Toto rozdělení kromě přehlednosti umožňuje jejich snadné použití při vývoji dalších zařízení.

4.5.1 Modul init

Jedná se o modul, který je automaticky po zapnutí zařízení vyhledán a spuštěn pomocí Lua interpretru. Z těchto důvodů nesmí být zkompilovaný do tvz. bytecodu. Modul init obsahuje implementaci všech ostatních modulů a spuštění iniciační funkce.

4.5.2 Modul config

Modul config obsahuje veškerou potřebnou konfiguraci - WiFi sítí, MQTT broakeru, GPIO pinů, nadmořské výšky pro přepočet tlaku, korekci měřených hodnot. Hodnoty jsou uloženy do jednotlivých proměnných, které po implementaci tohoto modulu jsou uchovávány v paměti RAM a jsou dostupné funkcím v dalších modulech. Kromě proměnných obsahující konfiguraci se v tomto modulu nachází dvě funkce, které slouží pro uložení a načtení nastavené teploty vytápění do souboru a tím její uchování v případě restartu řídícího modulu IoT.

4.5.3 Modul initsetup

Modul initsetup obsahuje spouštěcí funkci, ve které jsou inicializovány veškeré hardwarové komponenty, navázáno spojení s WiFi sítí a MQTT broakerem, načtena požadovaná teplota vytápění ze souboru uloženého v paměti flash a je nastavený časovač, který periodicky volá funkci pro aktualizaci. Funkce pro aktualizaci, která je periodicky vykonávána, provádí odečet hodnot ze senzoru a AD převodníku, vypsání nových hodnot na displej, kontrolu připojení k WiFi síti a k MQTT broakeru, odesílání aktuálních hodnot na broaker. Kromě těchto zmíněných dvou funkcí tento modul obsahuje funkci pro řízení vytápění, na základě porovnání změřené a požadované teploty. Aby nedocházelo k častému spínání vytápěcího systému a jeho opotřebení, případně i k poničení, je v této funkci implementována hystereze.

4.5.4 Modul connectWifi

Pro připojení k WiFi síti slouží modul connectWifi. Funkce v něm obsažené nejdříve zjistí dostupné WiFi sítě v okolí. Následně jejich seznam porovnají se seznamem známých WiFi sítí uložených v modulu config. Jakmile je nalezena shoda se známou

WiFi sítí, řídící modul IoT se k ní připojí. Veškerá síťová konfigurace kromě DNS serveru je převzatá z DHCP serveru.

4.5.5 Modul mqttConnection

Cílem modulu mqttConnection je navázání spojení a zajištění veškeré komunikace s MQTT broakerem. Z těchto důvodů obsahuje funkce pro navázání spojení s MQTT broakerem, registraci odebíraných témat, zpracování přijatých zpráv a odesílání aktuálních hodnot.

4.5.6 Modul display

Modul display se skládá z funkcí pro inicializaci displeje, přípravy dat k zobrazení, jejich vykreslení a funkce pro přepnutí displeje do úsporného režimu. Displej je z důvodu prodloužení životnosti displeje a úspory energie automaticky do tohoto režimu přepnout po uplynutí času nastaveného modulu config od doby aktivace displeje nebo manipulace s enkodérem při aktivním displeji. V úsporném režimu není na displeji nic zobrazováno.

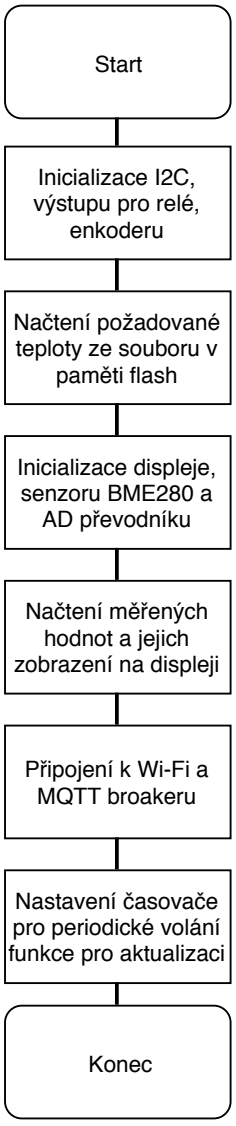
4.5.7 Modul measure

Modul measure se skládá z funkcí pro inicializaci senzorů BME280 a AD převodníku, odečet vstupního napětí (pro určení způsobu napájení) a naměřených hodnot senzorem. Dále tento modul obsahuje proměnné, do nichž jsou ukládány změřené hodnoty, které jsou následně zpracovány funkcemi v ostatních modulech. Při testování bylo zjištěno, že většina napájecích adaptérů neposkytuje při zatížení (nabíjení záložní baterie) konstantní napětí 5 V. Z těchto důvodů není možné určit způsob napájení pouze podle úrovně vstupního napětí a bylo zapotřebí implementovat funkci, která sleduje i jeho změnu.

4.6 Činnost firmware

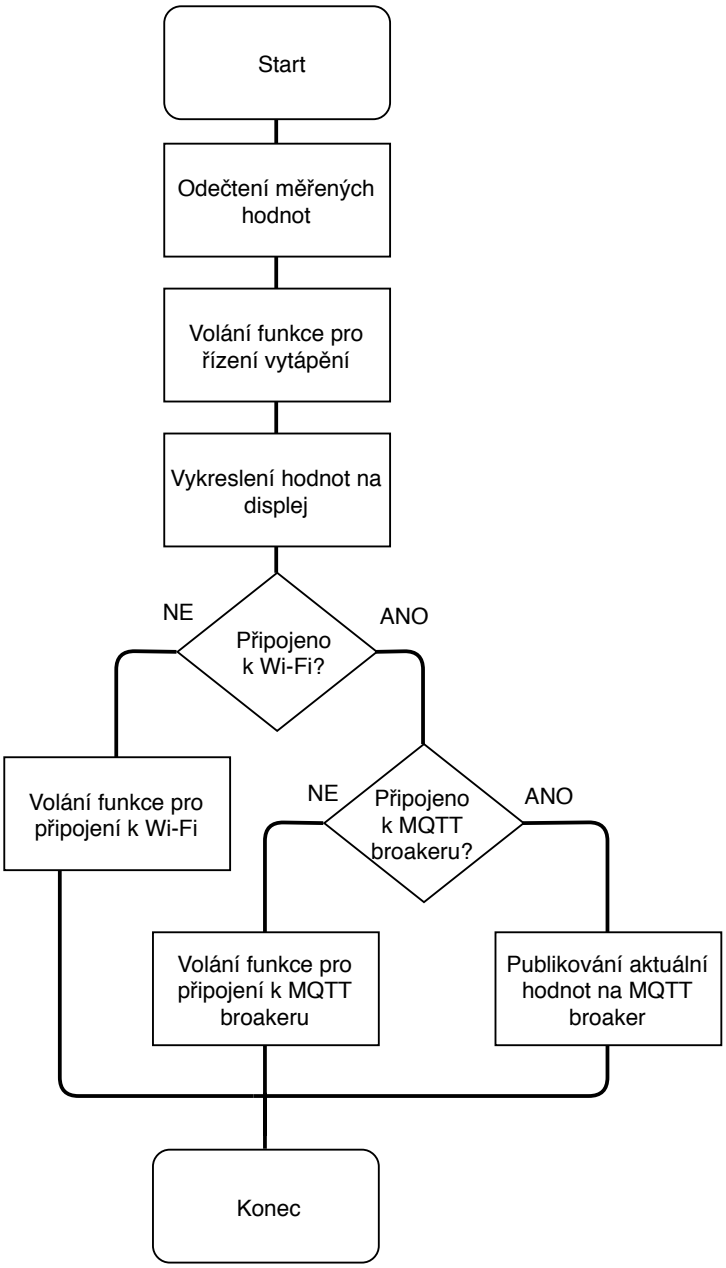
Činnost skriptu se dá rozdělit na dvě hlavní části – inicializace a aktualizace dat.

Inicializace (viz obr.4.5) má za úkol provést nastavení sběrnic, vstupních a výstupních pinů, načíst uloženou požadovanou teplotu, inicializovat veškeré připojené komponenty, navázání spojení s WiFi síti a MQTT brokerem. Na závěr inicializace je nastaven časovač, který periodicky volá funkci pro aktualizaci dat.



Obr. 4.5: Vývojový diagram funkce pro inicializaci.

Funkce pro aktualizaci dat (viz obr.4.6) provádí odečtení naměřených hodnot, následně volá funkci pro řízení vytápění a vykreslení aktuálních dat na displej. Poté zkontroluje připojení k WiFi síti a MQTT brokeru. Pokud je navázáno spojení s WiFi sítí a MQTT brokerem, jsou aktuální data odeslána na MQTT broker. V opačném případě jsou volány funkce pro obnovení spojení.



Obr. 4.6: Vývojový diagram funkce pro aktualizaci dat.

Pro zpracování náhlých událostí, jako je např. pootočení nebo stisk enkodéru, přijetí zprávy z MQTT broakeru, jsou automaticky po vzniku události volány funkce pro jejich zpracování. Tyto funkce jsou nastaveny při inicializaci daných knihoven.

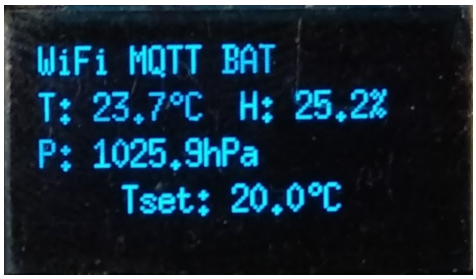
5 OVLÁDÁNÍ A ZOBRAZENÍ DAT

Ovládání navrhnutého řídícího modulu IoT a zobrazení aktuálních dat je možné dvěma způsoby - dálkově pomocí MQTT protokolu nebo fyzicky pomocí displeje a enkodéru, které se nachází na realizovaném řídícím IoT. Oba způsoby jsou popsány v následujících kapitolách.

5.1 Přímé ovládání a zobrazení dat na řídícím modulu IoT

Při návrhu způsobu ovládání byl kladen důraz na jednoduchost a možnost intuitivního pochopení bez nutnosti použití návodu.

Po stisku enkodéru je aktivován display (viz obr.5.1), na jehož prvním řádku je zobrazen stav připojení a způsob napájení. To je realizováno vypsáním označením daného stavu („WiFi“ - připojeno k WiFi síti, „MQTT“ - připojeno k MQTT, „USB“ - napájeno přes USB, „BAT“ - napájeno z baterie). Na dalších dvou řádcích jsou vypsány naměřené hodnoty prostředí (teplota, vlhkost a tlak). Na spodní části displeje se zobrazuje nastavená požadovaná teplota označená „Tset“, kterou lze změnit pootočením enkodéru při aktivním display. Displej se při nečinnosti po uplynutí nastavené doby automaticky vypne. Pokud dojde ke změně požadované teploty, je po vypnutí displeje tato hodnota uložena do flash paměti (čímž se eliminuje počet zápisů do této paměti) a je odeslána na MQTT broker.



Obr. 5.1: Zobrazení dat na displeji řídícího modulu IoT.

5.2 MQTT Dash

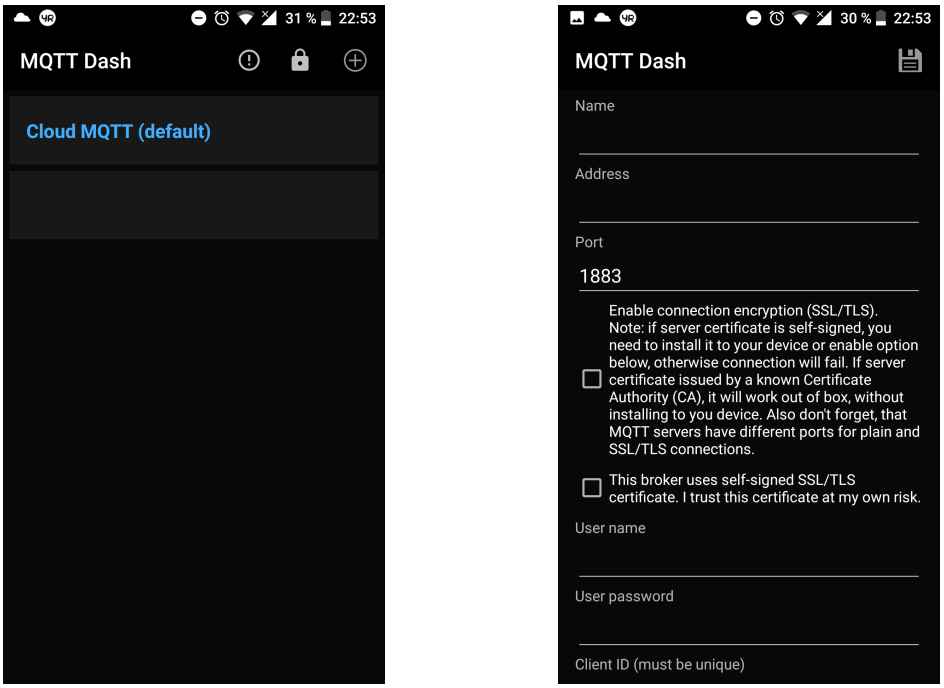
MQTT Dash je volně dostupná, bezplatná aplikace pro zařízení s operačním systémem Android pomocí internetového obchodu Google Play.[1] Tato aplikace je určena k přehlednému zobrazení dat z IoT zařízení a jejich ovládání pomocí MQTT protokolu (viz obr.5.2). Pomocí této aplikace byla ověřena možnost dálkového ovládání a zobrazení dat z navrženého řídícího modulu IoT.

Obr. 5.2: MQTT Dash - obrazovka s ovládacími a zobrazovacími prvky.

[1] Aplikace MQTT Dash je dostupná na stránce
https://play.google.com/store/apps/details?id=net.routix.mqttdash

5.2.1 Konfigurace připojení k MQTT broakeru

Po úspěšném nainstalování a spuštění aplikace lze pomocí stisku tlačítka se symbolem „+“ v pravé horní části okna přidat další konfiguraci spojení s MQTT brokerem. V konfiguračním okně (viz obr.5.3b) je nutné zadat název spojení, adresu MQTT broakeru, port, na kterém MQTT broker komunikuje, přihlašovací údaje pro autentizaci a unikátní označení klienta (zařízení s touto aplikací se připojuje k MQTT broakeru jako klient). Následně je nutné toto spojení uložit pomocí symbolu diskety, který se nachází v horní části displeje. Po uložení spojení je zobrazen seznam uloženými spojeními (viz obr.5.3a).



(a) Seznam uložených spojení s MQTT broakery.

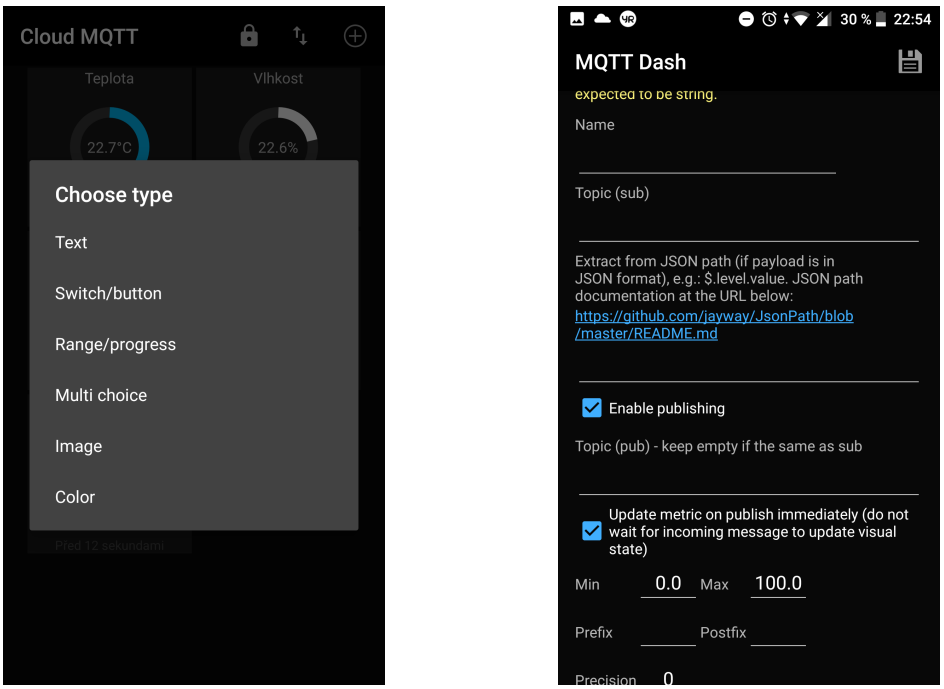
(b) Konfigurace spojení s MQTT broakerem.

Obr. 5.3: MQTT Dash - konfigurace připojení k MQTT broakeru.

5.2.2 Konfigurace ovládacích a zobrazovacích prvků

Jakmile je dokončeno nastavení spojení s MQTT brokerem je možné toto spojení vybrat ze seznamu. Po jeho výběru se zobrazí okno se zobrazovacími ovládacími prvky. Tyto prvky se přidávají pomocí symbolu „+“, po jehož stisku je zobrazena nabídka ovládacích a zobrazovacích prvků (viz obr.5.4a). Po výběru vhodného prvku se otevře konfiguračních okno tohoto prvku.

V tomto konfiguračním okně (viz obr.5.4b) je nutné nastavit název prvku, dané téma, kde se nachází hodnota pro tento prvek, rozsah této hodnoty, předponu před hodnotou a příponu (např. jednotka zobrazované hodnoty). V případě, že tento prvek slouží i k ovládání, je nutné zadat téma, do kterého má být odesílána nastavená hodnota, QoS a retain značka (určuje zda má MQTT broker tuto hodnotu uložit do doby, než bude nahrazena novou hodnotou). Jakmile je konfigurace dokončena, je ji nutné uložit pomocí symbolů diskety, která se nachází v horní části okna. Následně je opět zobrazeno okno se zobrazovacími a ovládacími prvky (viz obr.5.2).



(a) Nabídka ovládacích a zobrazovacích prvků.

(b) Konfigurace ovládacích a zobrazovacích prvků.

Obr. 5.4: MQTT Dash - konfigurace ovládacích a zobrazovacích prvků.

Po stisku symbolu zámku je možné aktivovat přemisťování těchto prvků v rámci okna s ovládacími prvky. Stejným způsobem je možné jej deaktivovat. Konfiguraci a rozmístění prvků je možné přenést na další zařízení s touto aplikací připojené ke stejnému MQTT brokeru, a to pomocí stisku symbolu dvou šipek v okně se zobrazovacími a ovládacími prvky (viz obr.5.2).

Měřená data jsou dělena do následující témat: [2]

- **<ENDPOINT>/temperature** - změřená teplota v „$°C$“
- **<ENDPOINT>/humidity** - změřená vlhkost v „$\%$“
- **<ENDPOINT>/preasure** - změřený tlak v „hPa“
- **<ENDPOINT>/voltage** - změřené vstupní napětí ve „V“
- **<ENDPOINT>/supply** - způsob napájení
- **<ENDPOINT>/settemp** - nastavená požadovaná teplota v „$°C$“

[2]Nadřazené téma označené „<ENDPOINT>“ je definované v modulu „config“ pro každý řídící modul IoT zvlášť

6 BROAKER

MQTT broker je hlavním, nepostradatelným, centrálním bodem pro komunikaci pomocí MQTT protokolu. Poté co se k němu jednotliví klienti (zařízení) připojí, jeho hlavním úkolem je přijímat zprávy od přihlášených klientů a následně je rozesílat odebírajícím klientům, případně tyto zprávy i ukládat do doby, než jsou nahrazeny novou zprávou v daném tématu. Možností realizace MQTT broakeru je mnoho. Některé z nich jsou popsané v následujících kapitolách, další způsoby jsou podobné.

6.1 Mosquito

Mosquitto je poměrně rozšířený open source software umožňující realizaci MQTT broakeru. [1] Mezi jeho výhody kromě toho, že se jedná o open source software, patří velké množství konfigurace a podporuje širokou škálu operačních systémů. Mezi podporované operační systémy patří Microsoft Windows, macOS, iOS a mnoho linuxových distribucí. Při provozu v domácnosti je kladen důraz na nízkou spotřebu energie platformy, která realizuje MQTT broaker. Vzhledem k podpoře OpenWrt (linuxová distribuce určena pro routery) a Raspberry Pi, lze nízké spotřeby dosáhnout provozem tohoto softwaru na routeru (užívajícího linuxové distribuce OpenWrt) nebo provozem na jednodeskovém počítači Raspberry Pi.

6.2 ClaudMQTT

ClaudMQTT je claudová webová služba, která realizuje MQTT broaker pomocí dříve zmíněného softwaru Mosquitto.[2] Po registraci si uživatel vybere a vytvoří instanci mosqutto. Instance mosquitto jsou de facto „balíčky“ poskytovaných služeb, které jsou různě omezeny a zpoplatněny měsíčním poplatkem. Po otevření vytvoření instance jsou vygenerovány a zobrazeny údaje nutné pro připojení klientů k takto vytvořenému MQTT brokeru. Hlavní výhodou této služby je, že není nutné provádět konfiguraci mosquitto softwaru a vlastnit platformu, na které by byl provozován.

Tento způsob realizace MQTT brokeru byl zvolen pro vývoj a testování vytvořeného řídícího modulu IoT z důvodu vyhnutí se chybám způsobeným špatným nastavením mosquitto softwaru. Pro tyto účely byla vybrána bezplatná instance mosquitto s názvem „Cute Cat“, která umožňuje připojení pěti klientů s přenosovou rychlostí 10 Kbit/s.

[1] Software Mosquitto je dostupný na stránce https://mosquitto.org/

[2] https://www.cloudmqtt.com/

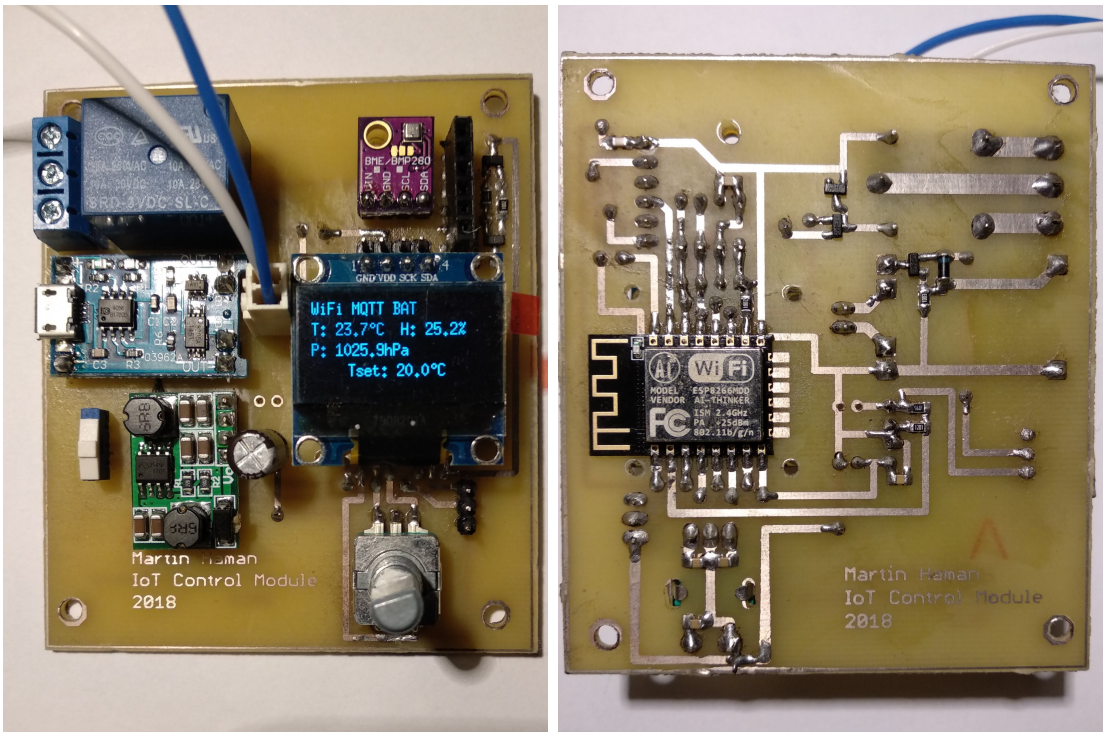
6.3 Zabezpečení přenášených zpráv

Data z IoT zařízení se mohou zdát na první pohled jako málo citlivá a nezneužitelná. Opak je pravdou. Tato data můžou být zneužita např. pro marketingové účely, zjištění přítomnosti osob v budově, neoprávněné ovládání nebo sabotáž. Toto může vést ke ztrátě důvěry uživatelů k IoT zařízení. Ze zmíněných důvodů je tedy důležité myslet na zabezpečení přenášených dat.

Kromě autentizace při připojení k MQTT broakeru pomocí přihlašovacích údajů lze přenášené zprávy zabezpečit pomocí kryptografických protokolů TLS a SSL. Dalším způsobem, jak zvýšit bezpečnost přenášených dat, je provozování vlastního MQTT broakeru v rámci lokální sítě oddělené od veřejné sítě pomocí firewallu. Připojení klientů nacházejících se mimo tuto fyzickou lokální síť lze realizovat např. pomocí VPN připojení.

7 REALIZACE A OVĚŘENÍ FUNKCIONALITY ŘÍDÍCÍHO MODULU IOT

Po osazení vyrobeného plošného spoje pro řídící modul IoT byl do něj nahrán zkompilovaný firmware NodeMCU (viz obr.7.1). Následně byl vytvořen a odladěn skript pro řízení vytápění.



(a) Vrchní strana. (b) Spodní strana.

Obr. 7.1: Osazený plošný spoj řídícího modulu IoT s nahraným firmwarem a skriptem pro řízení vytápění.

Funkčnost řídícího modulu IoT, možnost jeho ovládání a zobrazení dat byla úspěšně ověřena prostřednictvím obou možných způsobů - přímého manuálního ovládáni pomocí enkodéru a displeje i dálkového ovládání skrze MQTT protokol pomocí aplikace MQTT Dash (viz obr. 5.2 s aktuálními přijatými daty).

8 ZÁVĚR

V této bakalářské práci byla prostudována problematika přenosu dat z řídících modulů IoT. Byly popsány a porovnány vybrané bezdrátové technologie a aplikační protokoly vhodné pro přenos dat z IoT zařízení.

Hlavní vlastní přínos této práce spočívá v navržení, realizaci a ověření funkcionality řídícího modulu IoT. Vytvořený skript, který je v řídícím modulu IoT použit, umožňuje řízení vytápění budovy, měření teploty, tlaku a vlhkosti okolního vzduchu. Řídící modul IoT umožňuje ovládání a zobrazení naměřených hodnot přímo pomocí displeje a enkodéru nebo dálkově pomocí MQTT protokolu. Řídící modul IoT je zálohován pro případ výpadku síťového napájení, který je schopen detekovat. Vyrobené IoT zařízení je zamýšleno pro využití v domácnosti. Proto pro připojení k internetu byla zvolena bezdrátová technologie WiFi, která je již v dnešní době běžně používanou technologií a není tak zapotřebí dalších investic. Pro přenos dat byl vybrán MQTT protokol, jehož výhodou je malá hardwarová náročnost a to, že aktuální data jsou uchovávaná na jednom místě - serveru (MQTT brokeru). Tím pádem není nutné data získávat z jednotlivých IoT zařízení zvlášť.

Funkčnost navrženého a vyrobeného řídícího modulu IoT byla úspěšně ověřena pomocí manuálního ovládání zařízení, tak i dálkovým ovládáním z mobilního telefonu skrze aplikaci komunikující přes MQTT protokol.

Vyrobený řídící modul IoT lze dále po vytvoření a aplikování adekvátního skriptu použít např. pro řízení ventilace, zalévání, osvětlení, k základním meterologickým předpovědím apod. Jeho možnosti lze rozšiřovat díky vyvedené I2C sběrnici a pinu GPIO 16, který může být použit např. pro 1-wire sběrnici.

Oproti běžným spotřebičům IoT zařízení přináší novou funkcionalitu, což povede k zjednodušení lidské činnosti a také k jejímu zefektivnění. Je však nutné důsledně dbát na zabezpečení komunikace s nimi.

LITERATURA

[1] Internet věcí. POHANKA, Pavel. *Pavel Pohanka* [online]. [cit. 2017-12-07]. Dostupné z: http://i2ot.eu/internet-of-things/

[2] WiFi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-10-29]. Dostupné z: https://cs.wikipedia.org/wiki/Wi-Fi

[3] Typical WiFi Network. In: *LifeNet* [online]. Atlanta: LifeNet, c2011 [cit. 2017-10-30]. Dostupné z: http://www.thelifenetwork.org/images/typ_wifi.png

[4] What is LoRaWAN. In: *LoRa Alliance* [online]. Beaverton: LoRa Alliance [cit. 2017-10-22]. Dostupné z: https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf

[5] LoRaWAN 101 — A Technical Introduction. In: *LoRa Alliance* [online]. Beaverton: LoRa Alliance [cit. 2017-10-22]. Dostupné z: https://docs.wixstatic.com/ugd/eccc1a_20fe760334f84a9788c5b11820281bd0.pdf

[6] LoRaWAN. *IoT portál* [online]. IoT portál [cit. 2017-10-22]. Dostupné z: https://www.iot-portal.cz/2016/02/29/lorawan/

[7] LoRa network architecture. In: *Embedded Experience* [online]. Semtech [cit. 2017-10-23]. Dostupné z: http://2.bp.blogspot.com/-pBV2PL8CSAQ/VdoNG-I1gzI/AAAAAAAAAc4/mcfqb5P83Z0/s1600/LoRa-network-architecture-Semtech.jpg

[8] Z-Wave. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-11-04]. Dostupné z: https://en.wikipedia.org/wiki/Z-Wave

[9] Sigfox Technical Overview. In: *Sigfox* [online]. Labège: Sigfox, 2017 [cit. 2017-11-04]. Dostupné z: https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf

[10] ZigBee. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-11-08]. Dostupné z: https://cs.wikipedia.org/wiki/ZigBee

[11] ZigBee topologie sítě. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-11-08]. Dostupné z: https://cs.wikipedia.org/wiki/ZigBee#/media/File:Topologie_siti.jpg

[12] Hypertext Transfer Protocol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-12-07]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[13] Constrained Application Protocol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-12-07]. Dostupné z: https://en.wikipedia.org/wiki/Constrained_Application_Protocol

[14] A History of MQTT — Protocols for IIoT Applications. In: *InduSoft Web Studio* [online]. 2016 [cit. 2017-11-13]. Dostupné z: http://www.indusoft.com/blog/2016/09/23/a-history-of-mqtt-protocols-for-iiot-applications/

[15] MALÝ, Martin. Protokol MQTT: komunikační standard pro IoT. In: *Root* [online]. Root, 2016 [cit. 2017-11-13]. Dostupné z: https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/

[16] MQTT. In: *PAGE FAULT BLOG* [online]. [cit. 2017-11-15]. Dostupné z: https://pagefaultblog.files.wordpress.com/2017/02/mqtt.png?w=640

[17] ESP-12F specification. In: *Ai-thinker* [online]. Ai-thinker, 2017 [cit. 2017-12-07]. Dostupné z: http://wiki.ai-thinker.com/_media/esp8266/docs/aithinker_esp_12f_datasheet_en.pdf

[18] ESP-12F. In: *Banggood* [online]. [cit. 2017-12-07]. Dostupné z: https://img.banggood.com/thumb/view/oaupload/banggood/images/B3/D9/0861edf1-3c60-410a-924c-48e5cdc40bee.JPG

[19] ESP8266EX Datasheet. In: *Espressif* [online]. Espressif, 2017 [cit. 2017-12-07]. Dostupné z: http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[20] ESP8266EX functional block diagram. In: *IoT Bbits* [online]. Espressif, 2017 [cit. 2017-11-29]. Dostupné z: http://iot-bits.com/documentation/introduction-esp8266-native-development/

[21] 0.96"I2C OLED displej. In: *Ebay* [online]. [cit. 2017-12-07]. Dostupné z: https://www.ebay.com/itm/0-96-I2C-IIC-SPI-Serial-128X64-White-OLED-LCD-LED-Display-Module-for-Arduino-/201428440360

[22] BME280 modul. In: *Instructables* [online]. [cit. 2017-12-07]. Dostupné z: https://cdn.instructables.com/F0P/2YXR/J2MANTRV/F0P2YXRJ2MANTRV.MEDIUM.jpg

[23] TP4056 datasheet. In: *Amazon CloudFront* [online]. NanJing Top Power ASIC [cit. 2017-12-07]. Dostupné z: https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

[24] Auto Buck-Boost DC DC Converter Voltage regulator module 0.9-6V to 3.3V. In: *AliExpress* [online]. [cit. 2017-11-19]. Dostupné z: https://ae01.alicdn.com/kf/HTB1B5QXNVXXXXbJXXXXq6xXFXXXE/2PCS-Auto-Buck-Boost-DC-DC-Converter-Voltage-regulator-module-0-9-6V-to-3-3V.jpg

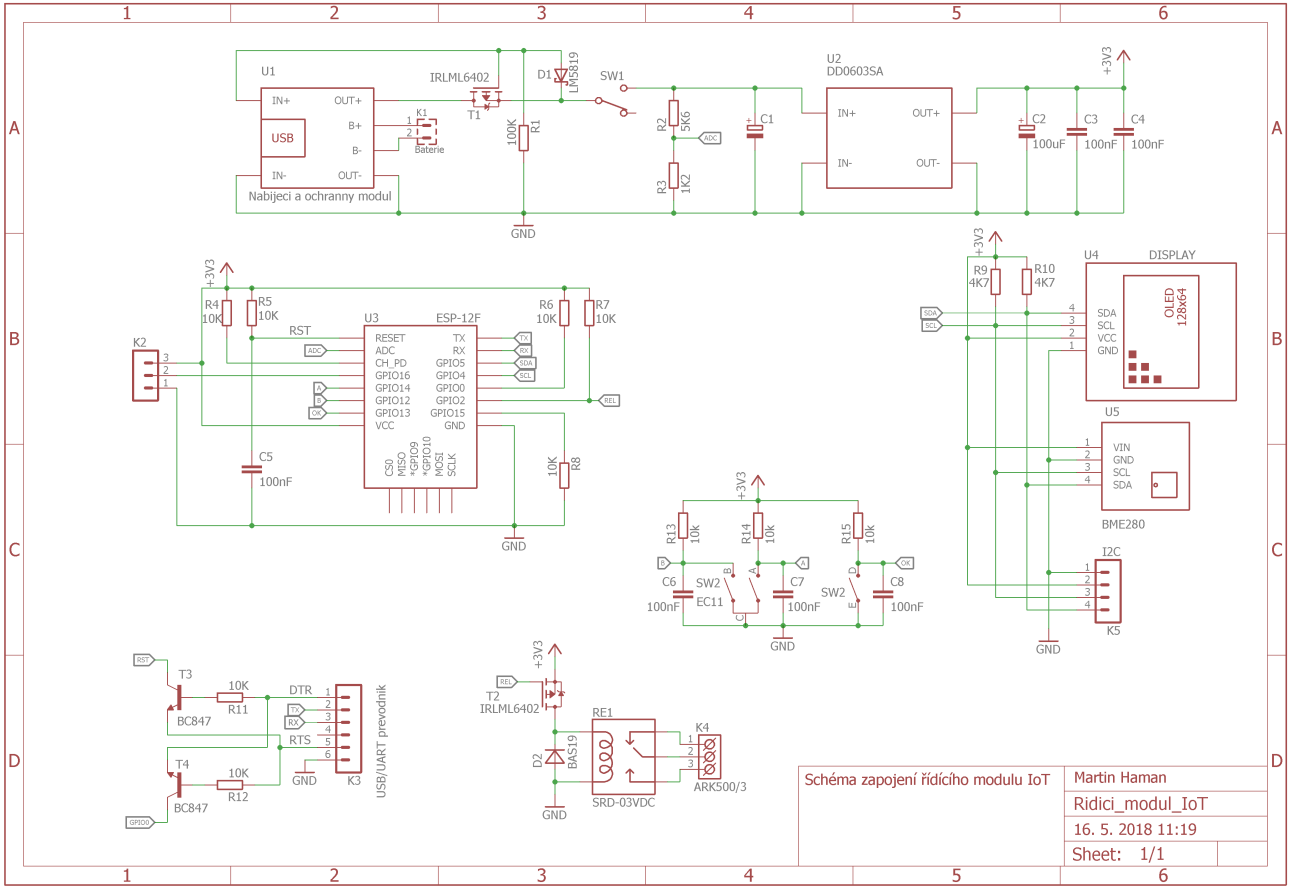
[25] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AT89C2051: [měření, řízení a regulace pomocí několika jednoduchých přípravků]*. Praha: BEN - technická literatura, 2006, 320 s. ISBN 80-730-0174-8.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AES	standard pokročilého šifrování – Advanced Encryption Standard
AP	přístupový bod – Access Point
CoAP	Constrained Application Protocol
DSSS	technika přímého rozprostřeného spektra – Direct Sequence Spread Spectrum
FFD	zařízení s plnou funkčností – Full Functional Device
GPIO	General Purpose Input/Output
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	„institut pro elektrotechnické a elektronické inženýrství“ – Institute of Electrical and Electronics Engineers
IoT	internet věcí – Internet of Things
IP	Internet Protocol
LAN	lokální síť – Local Area Network
LoRa	velký dosah (standart definující rádiovou modulaci) – Long Range
LoRaWAN	komunikační protokol využívající LoRa
LPWAN	Nízkoenergetická globální síť – Low Power Wide Area Network
MAC	unikátní identifikátor síťového zařízení – Media Access Control
MCU	jednočipový počítač (mikroprocesor) – Microcontroller Unit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MQTT	Queuing Telemetry Transport
NB-IoT	úzkopásmový internet věcí – Narrowband – Internet of Things
NC	běžně sepnuto – Normally Close
NO	běžně rozepnuto – Normally Open
QoS	kvalita služeb (přenosu) – Quality of Service
QPSK	digitální modulace založená na kvadraturním klíčování fázovým posuvem – Quadrature phase-shift keying
RFD	zařízení s omezenou funkčností – Reduced Functionality Device
SDK	soubor nástrojů pro vývoj softwaru – Software development kit
SoC	integrovaný obvod zahrnující všechny potřebné součásti – System on Chip
SSID	identifikátor bezdrátové sítě WiFi – Service Set Identifier
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WEP	Wired Equivalent Privacy
WiFi	"bezdrátová věrnost"– wireless fidelity
WPA	Wi-Fi Protected Access

SEZNAM PŘÍLOH

A SCHÉMA ZAPOJENÍ NAVRŽENÉHO ŘÍDÍCÍHO MODULU IOT



Obr. A.1: Schéma zapojení navrženého řídícího modulu IoT.

B OBSAH PŘILOŽENÉHO CD

- `/` ..... kořenový adresář přiloženého CD
  - `firmware` ..... zkompilovaný firmware NodeMCU
    - `nodemcu-master-18-modules-2018-03-08-15-46-32-float.bin`
    - `nodemcu-master-18-modules-2018-03-08-15-46-32-integer.bin`
  - `skript` ..... vytvořený skript rozdělený do jednotlivých modulů
    - `config.lua`
    - `connectWifi.lua`
    - `display.lua`
    - `init.lua`
    - `initsetup.lua`
    - `measure.lua`
    - `mqttConection.lua`
  - `schema-PCB` ..... Návrh schématu a PCB v Autodesk Eagle
    - `Ridici-modul-IoT.sch`
    - `Ridici-modul-IoT.brd`